# SUPER UTILITY 4/4P

FOR THE TRS-80 MODEL 4/4P

This software is made available on an **AS-IS** basis only, and Breeze/QSD, Inc shall not be liable for any actual or consequential damages, whether real or alleged, arising from the use of this software. No warranties, expressed or implied, are given with regard to the suitability of this product for any particular purpose. Breeze/QSD, Inc. warrants the product to do what it has been advertised to do; however determination of suitability for any purpose whatsoever is the sole responsibility of the end user.

A PowerSOFT Product from
Breeze/QSD, Inc.
11500 Stemmons Expressway, Suite 125
Dallas TX 75229
(214) 484-2976



Manual printing history:
First printing April 1984

Acknowledgements:

# TABLE OF CONTENTS

Table of contents, cont'd.

Table of contents, cont'd.

## PREFACE
### Background and Admonition

Super Utility Plus has a long history of adapting itself to the changing conditions in the TRS-80 world. The original version of the program, designed for the Model I, adapted itself to the advent of double density; then came the Model III, and Super Utility Plus moved to handle that particular computer too. There is even a variant for the Lobo MAX-80 computer, which is a TRS-80 "workalike" computer. Super Utility Plus even kept up with the advent of newer and more complex operating systems capable of writing different disk formats.

The TRS-80 world continues to change; new computers, new operating systems, new software continues to appear. And inevitably, their appearance is followed by a rising demand for a new version of Super Utility Plus capable of handling the new changes. The requests and demands NEVER go unheeded at PowerSOFT; you are holding the proof in your hands right now: Super Utility 4.

Super Utility 4 is written specifically for the Radio Shack TRS-80™ Model 4/4P computer. The Model 4/4P is the first TRS-80 to have an all-RAM mode, in which all of memory is available for program use, rather than having the first 16K of address space devoted to a fixed ROM (Read-Only Memory). For programs which do not make use of any routines in the BASIC ROMs, this arrangement frees up an additional 16K of memory for use. Super Utility Plus, having been written from the start to be as independent of the ROM routines as possible, is such a program. This is a godsend, because the most recent version of Super Utility Plus for the Models I and III (version 3.2) was running desperately short of buffer space. Even though double-sided operations had been implemented in that version, it was not possible to copy files to or from double-sided disks because of the lack of buffer memory. This version of Super Utility does not have that limitation.

Old SU+ hands will notice other changes in this version. Super Utility 4 removes a great deal of redundant code and other little-used routines. For example, the tape utilities are not present in this version; neither are routines such as Jump to Memory and Memory to Track, two routines whose use has become questionable because of the expanded use of memory (for buffer space) in Super Utility 4. The configuration display has also been simplified, and a new DOS specifier, T4D, has been added to handle TRSDOS 6.x disks (please note that CP/M is not supported).

This manual will describe in detail the operation of the program. Old users of Super Utility Plus will find themselves in familiar territory here; however we recommend a thorough reading of the manual before using the program for the first time. Things **have** changed from previous versions. If you are new to Super Utility 4, you are urged to read the manual carefully before using the program, and to keep it handy by your computer.

All users are urged to pay particular attention to the initial chapter, as it contains information necessary to the proper use of Super Utility 4, and later chapters will assume that you have read the first one.

The first chapter includes an overview of diskette data structure as it pertains to standard TRS-80 systems. The overview is not meant to be a tutorial on the organization of disk storage systems. If you desire in-depth information on this subject, there are other publications available which cover the it in detail. The classic is, of course, William Barden's **TRS-80 Disk Interfacing Guide**. For information on how directories are structured, Harvard Pennington's **TRS-80 Disk and Other Mysteries** is fast becoming a "standard" reference. Another book in the same series, Michael J. Wagner's **Machine Language Disk I/O and Other Mysteries**, addresses the subject of how to talk to the disk controller chip and make it do something meaningful. Still another useful publication is **Inside Super Utility Plus 3.0** by Paul Wiener and Gary Camp,

which provides much useful information on how to get Super Utility 4 to do various things to your disks.

The final section of the first chapter lists the symptoms of some of the most common disk I/O problems which make owning a copy of Super Utility Plus so vitally necessary for some people, along with their possible causes.

Super Utility 4 requires a minimum hardware configuration of a TRS-80 Model 4 or 4P with 64K of RAM. The additional 64K bank is not required, and will not be used. At least one disk drive is required, although the operation of the program will be considerably enhanced when two or more disk drives are available. Please note that this program is designed to work with standard Model 4 or 4P computers ONLY. Correct operation with computers which have had unofficial or non-Radio Shack modifications installed is ABSOLUTELY **NOT GUARANTEED**.

The Super Utility 4 diskette is designed to boot up on a Model 4 or 4P computer. If a Model 4P is being used, it will NOT be necessary to load the MODELA/III file prior to bootup. Simply place the SU+ diskette in drive 0 and press the RESET button. Super Utility 4 will operate the computer in its all-RAM mode.

Super Utility 4 does not require any external DOS support while it is running; in fact, it does not even need the presence of its own disk in drive 0. So after Super Utility 4 has loaded and the main menu is on the screen, you are strongly urged to open drive 0, extract the Super Utility 4 disk, and **PUT IT AWAY**. The less time it stays in the drive, the better its chances are against being damaged or destroyed by accident.

## REGISTRATION AND TECHNICAL SUPPORT

Please fill out the enclosed registration card completely and mail within 30 days of purchase to:

**PowerSOFT Products**
**11500 Stemmons Expressway, Suite 125**
**Dallas TX 75229**

Registration of this product in your name will entitle you to received technical support. Registered owners will be placed in the PowerSOFT customer database and will be informed of any upgrades, enhancements, fixes, or new releases of this program. Owners who do not register their copies will not be entitled to send their disks back for upgrade or receive any technical support.

If you are not sure whether or not to send your disks back for updating, please do the following:

1. Boot up your Super Utility 4 disk.
2. From the main menu, hold down the CLEAR and V keys of your computer simultaneously.
3. Note the version number, assembly date and revision number which appears on the screen.
4. Contact PowerSOFT Technical Support and ask what the latest version and revision numbers of Super Utility 4 are. Compare them with what you have. If they are the same, then there is no need to return your disks for updating.
5. If they are not the same, you have the choice of either sending your disk back for an update (a fee will be required) or installing the update yourself. If you wish to install it yourself, include a self-addressed stamped envelope and request the patch or update information.

Super Utility 4, to the best of our knowledge, contains no major bugs, but it is entirely possible that some minor ones may exist somewhere in the 35K+ of code that make up this program. If you encounter a problem, please re-read the

pertinent section of the manual first and determine whether or not the problem is real. See if it occurs on another computer. If you are unable to overcome it, please verify the exact circumstances under which it occurs and send a complete description to PowerSOFT Technical Support at the above address. Describe the problem as thoroughly as possible ("It doesn't work" does not tell us a whole lot) and always include a complete description of your hardware configuration (computer type [4 or 4P], number and capacity of disk drives, non-standard modifications, etc.). Also include your Super Utility 4 **SERIAL NUMBER**. We will do our best to assist all registered owners and answer queries in a timely manner.

If you have any urgent questions regarding the operation of this program, you may call PowerSOFT Technical Support at 214/484-2976 between the hours of 10 a.m. and 5 p.m. CENTRAL TIME (i.e., between 11 a.m. and 6 p.m. EASTERN, between 9 a.m. and 4 p.m. MOUNTAIN, or between 8 a.m. and 3 p.m. PACIFIC) Mondays through Fridays. If you are going to call please have your serial number on hand. You will be asked for it, and you must supply it and be verified in the PowerSOFT customer database before you can receive any technical support. It would also help if you could be at your computer with Super Utility 4 booted up and running when you call; many problems can be solved by just going through the procedure step by step under expert guidance.

If either of the supplied disks fails to load or operate properly, please return it to PowerSOFT for direct replacement within 10 days of purchase. Include a thorough explanation of the problem and a copy of your invoice showing date of purchase. If you have not yet sent in the registration card, you can do so at the same time. **Please note that if the problem is found to result from the use of non-standard hardware which PowerSOFT does not support, your disks will be returned to you unchanged.**

If your disks become unusable for any reason (e.g., you left it in the disk drive and then invoked "Software bulk

erase" on it while your dog was chewing up the backup copy) you may send them back to PowerSOFT for maintenance and repairs. Please check with PowerSOFT beforehand for the required maintenance fee. If your disks are found to be in unusable condition (e.g., teeth marks on them) they will be replaced and you will be charged the cost of a new disk.

When returning disks back to PowerSOFT, please package them securely between two pieces of stiff cardboard or other Post Office-proof material. Mark the package appropriately (e.g., "Handle with Care, Magnetic Media Enclosed", etc.).

## NOTICE

The information in this manual regarding the abilities of Super Utility 4 is current as of April, 1984. It is possible that in future, changes to the Model 4/4P computer or associated software may render all or part of Super Utility 4 unusable. Breeze/QSD makes no warranties as to the fitness or usability of this program with respect to future releases of the Tandy TRS-80 Model 4 or 4P computer, or of its TRSDOS operating system, or of any associated system, utility, or applications software. Users are urged to report any incompatibilities to PowerSOFT Technical Support at the address shown above so that they may be dealt with in a timely manner.

## Very Important!

Please do not forget to send in your registration card! This will entitle you to:

1) Updates as they become available.
2) Help and assistance through the mail or on the phone.
3) Regular mailings on new information, patches, etc.
4) Ongoing support...

No card... no support! There will be no exceptions to this rule. If you want us in the future, send in your card now!

# CHAPTER 1 - BASIC INFORMATION

Super Utility 4 is an extemely versatile and powerful utility package for use on the TRS-80 Model 4 and 4P computers. It is a direct descendant of Super Utility Plus Version 3.2 which is in use on Model Is, IIIs, and Lobo Syseems MAX-80s. Its many routines allow you, the user, to per form a great variety of tasks, ranging from direct examination and modification of of the contents of a diskette to restoring an unreadable diskette to a usable condition. With Super Utility 4 you may format diskettes in a variety of ways, in cluding mixed-density tracks; backup a diskette to another using a very fast routine; examine a particular file on a diskette; modify the contents of your computer's memory; and more.

This manual will explain the various utility routines available in Super Utility 4. Please read it carefully before attempting to use the program for the first time. Due to its power and versatility, Super Utility 4 can do great damage to your diskettes if carelessly used.

It is assumed in this manual that you are familiar with your disk operating system's features. More informaton on this may be found in the manual which came with your DOS. It is also assumed that you have at least a basic idea of how data is structured and kept on diskettes. This manual is NOT a tutorial on data structures or disk I/O techniques.

Although Super Utility 4 is designed to work with the Model 4/4P, you will have the ability to operate on other Model I and Model III diskette formats, **with the exception of CP/M**. Because of the lack of standardization in 5.25" CP/M formats, it is impossible to provide support for that system.

## OVERVIEW OF SUPER UTILITY 4

These are the utilities available to you in the program:

```
** Super Utility ! ** PowerSOFT Prod. ** By: Kim Watt **
  Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved

__ Utility Features __

 1 _ Zap Utilities          6 _ Memory Utilities
 2 _ Purge Utilities        7 _ File Utilities
 3 _ Format Utilities       8 _ Configuration
 4 _ Backup Utilities       9 _ Exit Program
 5 _ Repair Utilities

Choice? #__
```

Each selection is preceded by a number and represents a group of procedures. By entering the appropriate number from your keyboard, you will be taken to a sub-menu which will allow you to select from the various procedures available in the group.

**ZAP Utilities** are routines which permit you to directly examine, modify and copy the contents of a diskette. The routines in this group will also permit you to search a disk for a particular occurrence of bytes or characters.

**PURGE Utilities** will allow you to quickly and easily remove or recover files from a system or data diskette as well as clean up your directory. You may optionally remove all

traces of a file from a directory, change the diskette name, and view the disk directory before and after making changes.

**Disk FORMAT Utilities** are routines for formatting a diskette in a variety of ways. You may format a disk in the standard manner, but also construct format tracks to your own specifications, even tracks with mixed single and double density sectors! You may reformat a diskette without losing any data previously written on it. There is even a routine which removes all traces of data from a diskette, in effect bulk-erasing it.

The **BACKUP Utilities** will perform standard or special backups of one disk onto another. The destination diskette may optionally be formatted before backup begins. The backup routines perform a full disk backup and permit several recovery options should errors be encountered on the source diskette.

The **REPAIR Utilities** will let you restore an unreadable disk directory or damaged boot sector to a usable condition, if at all possible. You may also recover files which were killed by Super Utility 4, and check the disk directory for any errors which may have crept in and which might result in the destruction of valuable data later on.

**MEMORY Utilities** will perform a variety of functions on the RAM memory of your computer. You may examine the contents of memory, move the contents of a segment of memory to another location, search memory for the occurrence of a particular string, read a port, send a byte to a port, and transfer a disk track into memory for examination.

**FILE Utilities** are similar to the ZAP utilities, except that they are file-oriented rather than disk- oriented. You may used them to view the contents of a file on a diskette without knowing exactly where on the diskette the file resides. You may also compare two files for differences, copy files from one disk to another, display the remaining free

space on a disk, display the location of files on a disk, create new files, and check the status of your disk drives.

The CONFIGURATION system allows you to tailor your Super Utility 4 to your own system. You can tell the program the characteristics of your lineprinter if you have one. You can tell the program how many disk drives you have and what kind of disks you expect to be using in each one. You can also optionally write-protect a particular drive so that no data on it can be inadvertently changed. The configuration parameters can be saved on the Super Utility 4 disk for automatic loading when the program is next rebooted.

## EXECUTING SUPER UTILITY 4

The Super Utility 4 program is supplied on a special self-booting diskette. Insert the diskette into your computer's drive 0 and press the RESET button. The Super Utility 4 logo will appear on the screen while your disk drive continues to run. After a few seconds the disk drive will stop, and the logo will begin to flash. At this point, press any key to bring up the MAIN MENU.

If the Super Utility 4 does not boot on your computer, try to isolate the problem first. There are a number of reasons why this should occur. Try booting the same disk on another Model 4 or 4P; if possible, try switching boot drives. If the disk will still not boot, then follow the directions in the previous section of this manual for obtaining a replacement.

Drive adjustments are more critical with Super Utility disks than with regular disks, due to special formatting. Disk drive motor speed should be no more than 1% away from 300 rpm. The optimum range is 299 to 301 rpm. The read/write heads should not be out of alignment. Motor speed and head alignment are critical factors.

You should be aware that during the entire loading process, Super Utility 4 is performing a memory test on your computer's RAM to ensure correct operation of the program. If the message "Memory Error" or "Mem Err" appears on the screen at any point, you should get your computer's memory tested and replaced if necessary. Make sure ALL banks are tested.

SU4 does not require the presence of a DOS disk in drive 0 at all. It does not even require the presence of its own disk in drive 0, except when configurations are being saved. Once you have successfully brought up the main menu, remove the SU4 disk from drive 0 and put it in a safe place. You should never keep the Super Utility 4 diskette in the drive any longer than absolutely necessary.

SU4 is menu-driven. Its various functions are accessed through a series of menus which appear on your screen. By keying in the appropriate number for the routine you want, you will activate that particular function, or bring up a further sub-menu. Simply pressing **ENTER** will default to the first selection on the displayed list. To exit a function at any time, press the **BREAK** key. BREAK will return you to the current menu. To return to the main menu at any time, press either **SHIFT-BREAK** or the **F1** key.

If you are prompted for additional input after making a menu selection, you may enter requested numeric input in a variety of number bases. Decimal is the default. You may enter hexadecimal values by appending **H** to the number (e.g., 1AH), octal values by appending the letter **O** or **Q** (e.g., 177O or 177Q), or binary values by appending the letter **B** to the number (e.g., 100111B). String input may be typed in directly, and lower case may be used at any time. **SHIFT-0** ("shift-zero") and **CAPS** act as reverse case toggles. SU4 comes up in reverse case, that is, normal keys are upper case, and shifted keys are lower case. Pressing either shift-0 or CAPS will place you in "typewriter" mode, where normal keys are lower case and shifted keys are upper case. You can toggle back and forth at any time.

When several inputs are requested on a single prompt, you may enter them separated from each other by commas or spaces. For example, the command strings "T1S,30,10" and "T1S 30 10" are completely equivalent.

In addition to the various keyboard functions, SU4 has a powerful screen printer built right into the program. Pressing SHIFT-CLEAR at any time after the main menu comes up wil cause whatever is on the screen to be sent to the lineprinter. if your lineprinter is capable of reproducing the TRS-80 block graphic character set you can configure SU4 to send them out; otherwise block graphic characters will be changed to "#" characters on printing.

When a display is scrolling on the screen, it may be temporarily frozen by pressing the spacebar. Pressing ENTER will cause the scrolling to resume.

In addition to the various key controls, four control combinations are active at any point in the program. These are:

F2 - Toggles a "sound" function. When first pressed, the Model 4's sound generator will beep, and from that point on, all "<KEY> to continue" prompts (prompts requesting a keypress) will be accompanied by a short "beep." Pressing F2 a second time will disable the sound function.

CLEAR-A - toggles the moving graphics characters on the display on or off.

CLEAR-V - Displays the Super Utility 4 version and revision numbers, along with the assembly date of your copy. The revision number will help you determine whether or not your SU4 is in need of an update or not. Revision numbers reflect the patches which have been installed in the program, so to determine whether or not your copy needs to be updated, hold down CLEAR-V and note down the revision number which is

displayed. Then contact PowerSOFT and inquire what the latest version and revision numbers are for Super Utility 4. Compare these with what you have. If they are the same, then you have the most current version.

CLEAR-S - displays the serial number of your Super Utility 4 disk.

The last three control combinations require that you hold the CLEAR key down and press the appropriate letter key. In addition, note that all four are also subject to automatic keyboard repeat. In the case of CLEAR-A, which has a toggle function, holding the keys down too long may cause them to toggle twice, with a net effect of zero. Hold them down a shorter time.

## CONFIGURING SUPER UTILITY 4

Super Utility 4 may be configured to your system very easily. In the main menu, you will see a selection that reads, "CONFIGURATION." If you press the number 8 on your keyboard, you will see a display that looks like this:

```
. ** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **
.  Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved
.  __ Configuration __
=>Dual=N Graphics=N Locase=Y Linefeeds=N Usr Key=8 Speed=Y
. +:0 T4D' Ptks= 40 Rtks= 40 Dir= 20 Stp=0 Rdly=2 Wdly=2 WP=N
. +:1 T4D' Ptks= 40 Rtks= 40 Dir= 20 Stp=0 Rdly=2 Wdly=2 WP=N
. +:2 T4D' Ptks= 40 Rtks= 40 Dir= 20 Stp=0 Rdly=2 Wdly=2 WP=N
. +:3 T4D' Ptks= 40 Rtks= 40 Dir= 20 Stp=0 Rdly=2 Wdly=2 WP=N
. ? #_____
```

There are several lines to the display, and you will see a little arrow to the left of the top line:

**DUAL=N GRAPHICS=N LOCASE=N LINEFEEDS=N, USR KEY=8,SPEED=Y**

An arrow indicates which line is being worked on, and you can bring it down to the next by simply pressing the ENTER key.

The first four items on this line pertain to your system's lineprinter and its characteristics. **DUAL** refers to whether or not you want dual output on or off; if you turn it on, everything that appears on your screen will be sent to the lineprinter simultaneously.

**GRAPHICS** refers to whether or not your lineprinter is capable of printing the TRS-80 block graphic character set. If this is turned on, then SU4 will send the graphics codes out to your lineprinter. If turned off, then the program will translate all graphic codes to "#" symbols when sending them to the lineprinter.

Some printers can generate the TRS-80 block graphic characters but do not use the same ASCII codes as the computer. An example of this is the popular EPSON MX-series printers. If you are using an EPSON MX printer, you may force the correct offset by entering **GRAPHICS-M** (for "MX"). This will cause SU4 to output the correct graphics codes. Note, however, that later MX models no longer have the TRS-80 block character codes in them, so that this setting could have an undesirable effect. Along with other printers they must be considered as being INCAPABLE of generating the TRS-80 block graphic character set.

**LOCASE** simply refers to whether or not your lineprinter can print lowercase characters or not. If turned on, SU4 will send all lowercase codes to the printer as is; otherwise they will be translated into their corresponding uppercase codes before being sent to the lineprinter.

**LINEFEEDS** tells SU4 if your lineprinter requires linefeeds after each carriage return to properly move the paper. Most Radio Shack lineprinters do this automatically, and normally this option would be left off.

The next item is **USR KEY** and refers to the **F3** key of the Model 4. This key can be programmed to take you to any of the Super Utility 4 sub-menus, just as pressing F1 will take you

to the main menu. As supplied, it is set up to take you to the configuration system. However, by entering a different value for it, you can go to any of the other sub-menus by simply pressing F3. The values entered are the same numbers as would be found on the MAIN menu. For example, if you wanted F3 to take you directly to the REPAIR UTILITIES sub-menu, which is number five on the main menu, you would enter 5 for the USR KEY value.

Once programmed, the USR KEY would act just as BREAK or SHIFT-BREAK or F1 would; that is, pressing it will abort any operation in progress and take you immediately to the defined sub-menu.

Do not use the F3 key to enter the program from the opening graphics logo, as this will result in an incomplete initialization (part of the configuration information will not be read in). This is the only restriction on the use of F3.

The final item on this first line is **SPEED**, which allows control of the Model 4's CPU speed for a variety of purposes. As supplied, SPEED is set to Y, which means "run at high speed" (i.e., 4 MHz). If set to N, then the CPU would be slowed down to the normal Model III CPU speed of about 2 MHz.

Related to speed, but not shown on the configuration line, are two additional parameters which you can use to define the machine code necessary to turn SPEED on and off. This optional facility is provided in the event that you have additional modifications to the Model 4 which permit you to control CPU speed even further. The two parameters are **O** (for "On") and **F** (for "oFf").

To use these two parameters, you must enter the necessary machine code in HEXADECIMAL representation. You may enter up to 8 hexadecimal bytes for each parameter. If you enter less than 8 hexadecimal bytes, SU4 will pad them out with 00 (NOP) bytes.

Suppose your speed control modification required the following code to toggle it on:

```
3E01    LD      A,01
D37F    OUT     (7FH),A
```

This series of instructions translates into the byte sequence 3E01D37F. To enter this, following the speed parameter, you would enter:

**O=3E01D37F**

The **O=** identifies this as the machine code sequence to turn your speed modification ON, and the bytes are entered in hex, with no spaces between digits.

Similarly, if the modification required the following machine code to turn in off:

```
3E00    LD      A,00
D37F    OUT     (7FH),A
```

then the code would be entered into SU4 as follows:

**F=3E00D37F**

The **F=** identifies the sequence as the code needed to turn the speed modification OFF. SU4 will now use the two entered code sequences instead of its own code whenever the SPEED parameter is changed.

To turn these options on or off, simply enter **Y** (yes) for on, or **N** (no) for off, for each printer option, the necessary sub-menu number for the USR KEY option, Y or N for SPEED and any necessary code for the two unseen parameters O and F. For example, suppose you wanted to set your printer options as follows: Dual mode on, no graphics, lowercase on, linefeeds off. You would enter on the command line,

? Y,N,Y,N

and press the ENTER key (the ? is the SU4 prompt character and should not be entered). Note that you only have to enter the answers (Y or N), but they must be in the correct relative position. In other words, each reply must be in the same position as the option on the line above it. As soon as you press ENTER, you will see the top line change to reflect your answers, and the arrow (->) will move down to the next line. Invalid answers will be rejected; no change will take place in the configuration tables beyond the last correct entry.

Suppose you did not want to change all of the options, but only one, and that one happened to be the second, or third, or fourth, in the list? You must enter ALL the replies up to that point. For example, you only want to turn lowercase on and leave the rest as is. You would enter N,N,Y where the first two replies duplicate the settings of the first two options, and the third switches lowercase on. You may press ENTER at this point; it is not necessary to enter the fourth reply, since that option is being left unchanged.

The next four lines in the configuration display refer to the disk drives in your system. SU4 will assume four drives present, even if you have less.

Let's take a look at the entry for drive 0:

**+:0 T4D' PTKS=40 RTKS=40 DIR=20 STP=0**
**RDLY=2 WDLY=2 WP=N**

To the left of the drive number is a plus sign. This means that SU4 will recognize that drive as being in the system. If it were a minus sign, then SU4 will assume that it is not in your setup and will refuse to do any I/O to it. You can change this by typing either a plus or a minus as the first character on the prompt line.

For example, if you had only two drives, 0 and 1, you

might want to remove drives 2 and 3 from the system. In this way, if you should inadvertently reference drive 2 or 3, you will be informed that the drives are not available in the system.

You may also specify an equals sign, "=", indicating that SU4 should operate this drive in SKIP MODE. This is used only when trying to read a 35 track or 40 track disk in an 80-track drive. If one of your drives is an 80-track drive and you wish to read a 40-track disk in it, set that drive to SKIP. SU4 will step the head twice for every track and enable you to read that drive. However, be careful about writing data to a 4 35 or 40 track disk in an 80 track drive, as the result may not be readable in a regular 40-track disk drive afterward.

T4D' is a DOS SPECIFIER. This tells SU4 what Disk Operating System formatted the disk in that drive. The default is T4D', meaning TRSDOS 6.x for the Model 4, double density, single-sided. You can read almost any Model I or Model III disk with SU4, with the exception of CP/M or CP/M Plus diskettes. The DOS specifiers recognized by SU4 are given in the table below. Note that if more than one item appears in the Model I or Model III column for a particular DOS type, these items are equivalent in meaning and may be used interchangeably.

## Table of DOS Specifiers

| DOS | Model I | Model III | Model 4 |
|---|---|---|---|
| TRSDOS | | | |
| Sgl. Dens. | T1,TS,T1S | invalid | TS, T4S |
| Dbl. Dens. | T1D | T3,TD,T3D | T4, T4D |
| LDOS | | | |
| Sgl. Dens. | L1,LS,L1S | L,LS | TS, T4S |
| Dbl. Dens | L1D (SOLE disk) (NOTE 1) | L3,LD,L3D | T4, T4D |
| DOSPLUS | | | |
| Sgl. Dens. | D1,DS,D1S | DS,D3S | TS, T4S |
| Dbl. Dens. | D1D(sys. disk) (NOTE 1) | D3,DD,D3D | T4, T4D (NOTE 4) |
| MULTIDOS | | | |
| Sgl. Dens. | M1,MS,M1S | MS,M3S | (NOTE 5) |
| Dbl. Dens. | M1D(sys. disk) | M3,MD,M3D (NOTE 1,2) | (NOTE 5) |
| NEWDOS/80 V2 | | | |
| Sgl. Dens. | N1,NS,N1S | NS,N3S | (NOTE 5) |
| Dbl. Dens. | N1DR | N3R,NDR,N3DR (NOTE 1,2,3) | (NOTE 5) |
| DBLDOS | | | |
| Dbl Dens. | B1R,BDR,B1DR | invalid | invalid |

**Note 1:** For Model I data disks which do not have the single density track 0, use one of the Mod III codes.

**Note 2:** Relative sectoring as required by NEWDOS/80 V2 double density diskettes is specified by appending the letter R to the specifier, e.g., N1DR. This is mandatory for correctly handling ND80 and DBLDOS disks in file and directory oriented operations. The R modifier is allowed only in MULTIDOS Model I double-density to produce the "P-density" disks. However, The R modifier should be left OFF when doing Model I standard backups (only).

**Note 3:** For NEWDOS/80 V2 diskettes with a normal Track 0, use one of the corresponding Model III specifiers.

**Note 4:** DOSPLUS IVa for the Model 4 is format-compatible with

TRSDOS 6.x in most respects.

Note 5: Where no Model 4 equivalents of the format exist, use one of the Model III specifiers.

The table above covers most of the available formats of TRS-80 diskettes. For those diskettes which use a relative track/sector formatting scheme such as NEWDOS/80 V2 double density, MULTIDOS P-density, and DBLDOS, the "R" modifier is used for all operations except Model I Standard Backup. Leave off the "R" when specifying one of these disks for a Model I backup operation.

The specifiers L3D (LDOS double density) and T4D (TRSDOS 6 double density) are identical in terms of formatting, so SU4 treats them equivalently. Because of this, when one or more drives are set to L3D, for example, and one of them is changed to T4D, the others will ALSO change to T4D and vice versa. This will not make any difference in terms of how those disks are treated by the program, and memory is conserved by not having to create all the code necessary to deal with a separate DOS specifier.

Limited support for double-sided media is provided for LDOS (TRSDOS 6) and DOSPLUS (In the event that new versions of MULTIDOS are released which conform to the double sided standard used by the other two systems, then support for them will be available). A double-sided disk is specified by appending a double-quote to the DOS Specifier, e.g., L3D" indicates a double-sided LDOS disk. Conversely, a single-quote or apostrophe appended to the DOS Specifier indicates a single-sided disk. There is no limitation on the use of double-sided disks in Super Utility 4.

The DOS Specifiers can also be used within the other routines of SU4 to override the configuration table setting. To do this, append the appropriate DOS Specifier to a drive number when prompted for one. For example, when prompted for drive, track and sector, you might enter,

**0M3D'-40,21,5**

By appending **M3D** to the drive number, you are telling SU4 that the disk in that drive is now a Model III double-density MULTIDOS disk. **-40** indicates that it is formatted for 40 tracks. Similarly, when prompted to supply a filespec, you may enter,

**MYFILE/BAS:3LD"-40**

to indicate that the disk in drive 3 where the file MYFILE/BAS is to be found, is a double-sided double-density LDOS disk. The use of the override system avoids the necessity of returning to the configuration module every time you want to change a DOS Specifier. You should be aware that the new setting is ENTERED into the configuration table by the override system, and will be in effect from that point on.

**PTKS-40** is the number of formatted physical tracks that SU4 expects to find on the diskette in drive 0. If you are going to put a diskette with a different track count in this drive, you should change this value.

**RTKS-40** refers to the number of relative tracks on the diskette. Normally this value is the same as PTKS. However, for those systems which use a disk relative formatting scheme such as NEWDOS/80 V2 and DBLDOS, this figure will differ from that of PTKS. **NOTE:** This value is **NOT** user-controllable. Do not enter this value when changing a drive configuration. It is automatically calculated from the PTKS value by SU4.

**DIR-20** tells you which track the diskette directory is located in. If a disk relative format is being used, this value should be the relative and not the physical track location of the directory. This value may be left as is; SU4 is fully capable of locating a directory track by itself.

**STP-0** refers to the track-to-track stepping rate of your disk drive's read/write heads. It is a coded value, and the

corresponding rates in milliseconds are as follows:

| | |
|---|---|
| 0 | 6 |
| 1 | 12 |
| 2 | 20 |
| 3 | 30 |

The Model 4's disk drives are capable of stepping at the fastest rate, 6 milliseconds. Some Model 4P disk drives can step at 6 but not at 12; however they can step at 20 and 30. As supplied, the value for STP is set to 0, which should be usable for all Radio Shack supplied drives on the Model 4 and 4P. If your drives came from another manufacturer, consult the technical specifications for its maximum stepping speed. Do not specify a step rate faster than the drive is capable of, or you will produce I/O errors which can ruin your disk. If in doubt, set the step rate to 3.

**RDLY=2** refers to the delay, in quarter-seconds, from the time a disk drive's motor comes on to the time when SU4 first attempts to read data from the diskette. If set to "4" there will be a one-second delay from motor-on to the first read attempt. As supplied, the delay value is set to 2, or one-half second. If you find that SU4 has trouble reading your disk for the first time, increase the delay value.

**WDLY=2**, conversely, controls the amount of delay from the time the drive motor comes on to the time SU4 first attempts to write to the disk. If set to 4, SU4 will delay one second before attempting a write operation. As supplied, it is set also to 2, or one-half second. Since disk writes are more critical than reads, this delay factor is controlled independently to ensure reliable disk I/O. Again, if errors occur as a result of bad writes to the disk, you may wish to increase the delay value.

Both RDLY and WDLY can take values from 1 to 4. A value of 0 (theoretically, no delay) is invalid and will be rejected.

**WP-N** is a "switch" which tells SU4 whether or not to "write-protect" your drive. If you specify "Y" to this option, SU4 will read any diskette in this drive, but will not write to it. This is functionally equivalent to placing a write-protect tab on your diskette.

To alter the settings for a disk drive, you must enter a series of replies to the prompt which appears at the bottom of the screen. These replies describe the characteristics of your disk drive. The settings for each option must be in the correct relative order as the options list shown on the configuration display. The settings above would have been described by the following:

**+T4D',40,20,0,2,2,N**

The "+" is optional and indicates that the drive should be active in the system. The rest of the string tells SU4 that the disk is a single-sided Model 4 TRSDOS 6.x disk (T4D') formatted for 40 tracks (40) with the directory on track 20 (20). The drive itself is capable of stepping at 6 msec. (0), and that a one-half second delay should be used for both read and write (2,2). The drive should not be write-protected (N). Note that the RTKS value is not entered. Also, if the side specifier is not entered (T4D instead of T4D'), it will automatically default to whatever it was previously set at.

If we were to change the specification to a single-density Model I disk, we might enter,

**T1S,35,17**

**T1S** indicates that the disk in that drive will be single-density TRSDOS 2.3, formatted for 35 tracks with a directory on track 17. The rest of the information is the same and does not have to be re-entered. In most cases, it is also not necessary to enter the directory track location as SU4 is capable of locating it automatically.

If you wish to take a drive out of the system, it is only necessary to enter the "-" symbol to disable that drive. The other option settings will obviously not matter.

In general, the only things that need to be configured are the lineprinter specifications, and for each drive the DOS type, track count, step rate, I/O delays and write-protect status. As supplied, SU4 assumes that all 4 drives (:0 to :3) are active in the system; to take one or more drives out of the system, simply enter a "-" for each one.

The directory location will be determined automatically when SU4 goes to read a diskette in that drive, and that location will be entered into the configuration table automatically. You will see that if you swap a disk of a different density or track count into a drive and read it, the configuration table will change to reflect the characteristics of the new disk.

If you are configuring your disk drives in an identical manner (e.g., all of one DOS type), you may simply configure the first drive for that type. Then, with the pointer on the next drive, enter on the prompt line,

<d

where "d" is the drive number whose characteristics you want copied into the current drive. For example, if you had configured drive 0 to T4D",40,20 etc. and you want the exact same configuration on drive 1, you would move the pointer down to the drive 1 configuration line and enter on the prompt line, <0. The configuration characteristics of drive 0 will be duplicated for drive 1 when ENTER is pressed.

Conversely, if you want to copy the configuration data of one drive TO another drive, position the => pointer to the source drive (the drive containing the settings you want copied), and enter a **>d** on the prompt line, where **d** is the drive you want the settings copied TO. So if you wanted the configuration of drive 0 copied into drive 3, for example, position the pointer at drive 0 and enter **>3**. The configuration settings of drive 0 will be duplicated into drive 3 as soon as ENTER is pressed.

Other modifiers may follow the ">d" and "<d" indicators. For example, you might enter on the prompt line,

**<0,L1S,80,40**

This would FIRST copy the configuration settings of drive 0 into the target drive, THEN it would alter the resulting specifications to indicate an 80-track single-density LDOS disks with a directory on track 40.

Note that if you are going to use these special commands "<" and ">" must be the first character on the prompt line. If either one is not the first character on the line, then the whole line will likely be rejected.

After you have configured SU4 to your specifications, press ENTER one more time. You will now be asked whether or not you wish to save the current configuration. If you reply "Y" then you will be asked to place the Super Utility 4 diskette in drive 0. Make sure there is no write-protect tab on the diskette. Press ENTER. Your new configuration will be written to disk, and will automatically be loaded the next time you boot up Super Utility 4.

## OVERVIEW OF DISKETTE DATA STRUCTURE

When you format a disk using the FORMAT utility of your disk operating system, information is written to the disk in concentric rings called cylinders. Most disk drives nowadays will format 40 cylinders, although some older ones can only format 35. There are also disk drives which can format up to 80 cylinders or more, and still others which can format both sides of a disk instead of just one. A Model 4 TRSDOS 6.x diskette is generally formatted for 40 cylinders, and is single-sided.

Each cylinder in turn is formatted into sectors. The number of sectors laid down on a cylinder depends on the diskette's density and whether or not it is double-sided. A Model 4 will generally format a diskette single-sided in double density (SSDD). A Model I TRS-80 will format a disk single-sided in single density (SSSD) unless it is equipped with a double-density modification kit.

A SSSD cylinder contains 10 sectors of 256 bytes each, while a SSDD cylinder contains 18 sectors of 256 bytes each. These values double for a double-sided disk. These are the actual counts of sectors laid down on a cylinder; however, what an operating system CALLS them is arbitrary. Most operating systems number the sectors on a SSDD cylinder from 0 to 17, while a few others number them from 1 to 18.

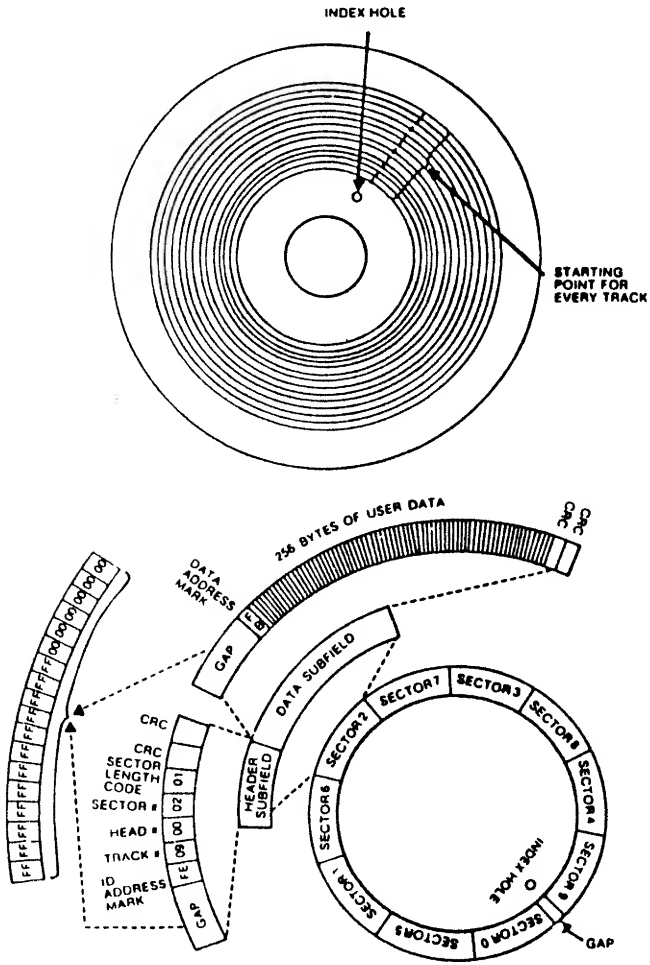
Whatever the density of the diskette, a sector always contains 256 bytes of data plus some additional bytes of information which identify the sector in terms of its cylinder number, sector number and side (sometimes referred to as "head") number. Each sector has two fields: the header field and the data field. The header field is a 6-byte field which consists of an address mark, a cylinder designator, a head designator, a sector number designator, a data length byte, and a two byte CRC (see below). Note that there is no separate field for a cylinder number. The cylinder number is contained

in the header field of every sector on the cylinder. The head number is used to identify which side of a double-sided diskette the sector is on. Some operating systems use it, others do not.

Following the sector number in the header field is a **length byte**. This is an encoded value of the number of data bytes contained in the following data field. There are two methods of encoding data length: IBM convention, and non-IBM (NBM) convention. TRS-80 diskettes usually use the IBM convention, in which a 0 corresponds to 128 bytes of data, 1 corresponds to 256, 2 refers to 512, and 3 refers to 1024 data bytes. Only values from 0 through 3 are used in double-density disks. The Non-IBM convention uses a 0 to indicate 1024 bytes per sector. Non-zero length bytes must be multiplied by 16 to arrive at the correct data length.

The header fields are placed on every cylinder of a diskette at format time, and are never altered unless the diskette is reformatted. The data fields, however, change each time the sector is written to by a program or the operating system. The data field contains the actual data plus 2 CRC bytes.

The **Cyclic Redundancy Check**, or CRC, bytes are the result of an extremely fast polynomial calculation performed by the disk controller chip on the data being written out to the disk (at format time for the header fields, and at sector write time for the data fields). When data is read back in from the disk, the CRC is recalculated as the data passes through the controller chip, and is compared with the two bytes placed on the disk. If they are the same, then the data was read in properly. If they are NOT the same, then the data was not read in correctly, and the operating system will usually try again. If, after a preset number of retries, the operating system is unable to get a good read, then an error condition is declared. This results in the messages, "Parity Error During Read," or "CRC error on disk read."




Data organization on a floppy diskette (single density)

Each field (both header and data) is preceded by a gap consisting of 18 bytes. These gaps are written to physically separate the sectors on the disk from each other. There is no firm convention established as to what the gap bytes should consist of, and they could be anything. Usually, however, they are made up of 12 bytes of FFH followed by 6 bytes of 00H.

The header and data fields on each sector are each preceded by an **address mark** which is used by the controller chip to detect the start of a block of data. The address marks are also used by TRS-80 disk operating systems to identify a cylinder as being part of a directory track or part of a data track. This is discussed further below.

The sectors on each cylinder are grouped together into arbitrary units called "granules," or **GRANS**. In TRS-80 parlance, a gran is the smallest amount of disk space the operating system will allocate to a file. As a file grows, additional grans are assigned to it as needed. The use of grans avoids excessive thrashing around of a drive's read/write head, such as what would happen if space was assigned on a purely sector-by-sector basis. It also speeds up disk access by ensuring that at least several sectors belonging to the same file will be side by side.

How big is a gran? On a single density disk, a gran consists of 5 sectors, thus a cylinder will be divided into two grans. A double density diskette formatted by TRSDOS 1.3 (Model III) has six grans of 3 sectors each on every cylinder. Model 4 TRSDOS (and most other double-density DOS) uses grans of 6 sectors in length, thus splitting up each cylinder into three grans. These values will double when double-sided disks are formatted, of course.

When a file is saved to disk, it may occupy several grans of space. It is the job of the disk operating system to keep track of where the file is located on the disk, and how much space it takes up. Sometimes a file must be broken up into several segments, or **extents**, in order to make maximum use of

available disk space. The operating system keeps track of all of this and maintains information on each file in a special place on the disk called the directory cylinder.

When you request a file, such as what happens when you issue the RUN "filename/ext" command from Disk BASIC, the operating system first goes to the diskette directory and looks for the file you specified. If it finds the file, the system then examines the information associated with that file to find out exactly where on the disk it resides, and how long it is. Armed with this information, the operating system then goes to the spot on the disk where the file begins and proceeds to load it into the computer's memory.

The directory cylinder is "marked" off from the rest of the cylinders of the diskette by the use of a special data address mark, or DAM. This is a piece of information written onto the disk by the operating system during the format process. This permits it to locate the directory cylinder quickly, as the directory DAM is different from address marks used on the other cylinders. However, the DAM written to the directory cylinder is dependent on the hardware in the computer, specifically, the disk controller chip.

The Model I TRS-80 uses a different controller chip from the Model III or Model 4. This poses certain problems, the chief one being the fact that you cannot read a Model I single density TRSDOS 2.3 diskette in a Model III or Model 4. The address marks on the directory cylinder are not readable by these two machines, and must first be changed before the disk can become readable. The problem of course is that once the address marks are changed, the disk essentially becomes unreadable to TRSDOS 2.3 on the Model I.

Reading a Model III or Model 4 disk on a Model I equipped with a double density adapter poses no problems, because the adapter uses both types of controller chips.

Many of the newer operating systems, including TRSDOS 6.x,

avoid this problem by using a directory DAM which both the Model I and Model III/4 disk controller chip can read and write.

The directory cylinder is one of the most important parts of a diskette. If the directory cylinder is damaged in some manner so that the operating system cannot read the information it contains, the diskette essentially becomes unusable. Any files stored on it are still there, but the operating system no longer has any way of knowing where exactly on the diskette each file is stored.

There is one other place which is vital to the operation of a system disk. This is the outermost, or boot, cylinder. When the reset button on a Model 4 is pressed, the first sector of this cylinder is read into a specific location in memory. This sector contains code which then allows the computer to read in the rest of the operating system. If the diskette is not a system diskette, then the boot sector code will produce the "Not a SYSTEM diskette" message on the screen. The other sectors of the boot cylinder may contain additional information pertinent to the system, such as configuration tables, the AUTO command string, etc. These additional sectors may be read by the operating system once it has been loaded into memory.

If the initial sector of a system disk's boot cylinder is damaged, that disk becomes unusable as a system diskette, although it may still be possible to use it as a data diskette. With Super Utility 4, it will be possible to restore diskettes which have sustained damage to the directory or boot cylinder, if the damage is not too extensive. Sometimes the damage is so widespread that no recovery is possible, but in many cases, a disk can be restored to working condition at least long enough for you to copy important files over to another diskette.

Many users who find that one or another of their system disks will not boot automatically assume that the boot

cylinder has been damaged. This may not necessarily be the case. It is important to realize that the process of bringing an operating system up to the point where it is ready to accept user commands consists of several steps. A failure at any one of these steps may give the impression that the boot cylinder has sustained damage. However, it may not be the boot cylinder at fault. It may be that the resident module of the operating system (SYS0) has been damaged, or is missing. It may also be that the command interpreter module (SYS1) is damaged or missing. Users who are unaware of this process might mistakenly assume that the failure is due to a bad boot cylinder, and that that is all which needs repairing. When the repair procedure fails to make the system function, they might get confused and assume that Super Utility 4 has developed a bug and malfunctioned. In fact it might have done EXACTLY what they wanted -- exactly, but no more.

When this happens, it pays to explore other possibilities before assuming that SU4 has malfunctioned. Try copying a good SYS0 or SYS1 module from a known good disk to the bad one. That may be all you need to get the system up and running again.

## TRACKING DOWN DISK DRIVE PROBLEMS

Super Utility 4 gives you a powerful set of tools with which to rescue glitched disks if necessary. However, it is important to understand why the disks got glitched in the first place. Did it happen by accident (e.g., were the disks placed near a magnet)? Did the damage occur outside the disk drive or inside? If inside, was it the result of a transient event or some consistent problem?

The answers to these questions are important. Super Utility 4 must work through the computer hardware, and if it was something in the hardware which damaged the disk in the first place, using Super Utility 4 can easily make matters worse unless the hardware problems are dealt with first.

The first place to check is the disk drive. Like any other piece of precision equipment disk drives need to be properly maintained for optimum performance.

Problems like "parity errors" or CRC errors can have a variety of causes. Below is a description of the most common ones.

**DRIVE MOTOR SPEED TOO FAST OR TOO SLOW.** 5.25" floppy drives turn at 300 rpm. Sooner or later, as a drive begins to age, the motor speed starts to drift from 300. However, there is a fairly wide tolerance to speed variations built into the drive electronics, and even though a drive's speed is no longer precisely at 300 rpm, no I/O errors occur. But if the speed drifts outside the allowable range, then errors start creeping in.

There are two speed ranges to be concerned about. One is the critical range for **formatting**, which is generally plus or minus 1 percent, i.e., 297 to 303 rpm. Drives whose motors turn within this range will not have any problems formatting a

disk or performing I/O to it. The second range is the range for standard disk I/O, which is generally plus or minus 3 percent of 300 -- that is, 291 to 309 rpm. Once a drive's motor speed falls outside this range, any type of I/O usually results in an error.

TRS-80 computers have a unique problem related to drive speed, and this has to do with the timing of the computer's "heartbeat" interrupt. This interrupt rate is almost the same as the rate at which a disk's index hole passes by the index sensor LED if a drive is turning at 300 rpm exactly. The result is usually a mysterious "hang" in which the disk drive remains selected but nothing appears to happen. The reason for this is the fact that just as the index hole of the disk passes the sensor, causing the drive to signal the controller that it has located the start of a cylinder, the interrupt fires, sending the computer off on some other task. By the time the computer returns, it is usually too late. This continues to happen till the two events drift out of synch.

TRS-80 owners who are aware of this will adjust their drives' motor speed away from 300 rpm, to prevent it from getting in synch with the heartbeat interrupt. The adjustment should be minimal -- no more than half a revolution away in either direction. Some people make the mistake of thinking that the larger the displacement, the better. While this may be true, it also hastens the time to when the motor speed drifts outside the allowable speed range, causing I/O errors. More is not always better.

**BAD HEAD ALIGNMENT.** A misaligned read/write head is a rare occurrence with the newer disk drives on the market today, unless they have been subjected to mechanical shock or other abuse. Head misalignment results in elliptical (egg-shaped) cylinders being written on the diskette. Alignment problems usually manifest themselves in disk I/O errors on the inner (higher numbered) cylinders, or in I/O errors when attempting to read or write a disk created in another drive. This can often be a subtle problem and very difficult to diagnose

because unless the alignment is very severe, a disk created on a misaligned drive will ALWAYS be readable on the same drive.

**WORN PRESSURE PAD.** This is a problem which has been responsible for more ulcers than it has been given credit for. On a single-sided drive, the **head load pressure pad** is a small piece of felt or plastic material which holds the diskette media against the read/write head. This is similar to the way the pressure pad in a regular tape cassette holds the tape against the recording head. Eventually the pressure pad gets worn down. When that happens, insufficient pressure is applied to the media, and disk I/O errors occur.

Some disk drives physically "unload" the head when the drive is deselected -- that is, the pressure pad is lifted away from the disk by a small spring. Other disk drives, however, do not load or unload the head -- the pressure pad is constantly in contact with the disk, resulting in faster wear. Drives of this type should always have the pressure pad checked when they are serviced.

**DIRTY READ/WRITE HEADS.** Just as a dirty recording head on a tape deck will produce distortions in music, so a dirty read/write head on a disk drive will produce distortions in data. The heads should be cleaned periodically with a cleaning disk. Always follow the manufacturer's instructions when using a cleaning disk. NEVER clean the drive head too often or you will wear them down. Unless your drives are in an exceptionally dirty environment, one cleaning every six months should suffice.

**LOOSE OR DAMAGED CABLES.** A disk drive cable which comes even slightly loose at either end can give rise to a multitude of disk errors. Make sure the connectors are always fastened tightly. If you suspect one or more strands of the ribbon cable have broken, replace the cable.

MEDIA WEAR. Although we would like to them to last forever, diskettes have a limited "lifespan." Generally a diskette will last for about 65 to 75 hours of constant head contact. Some premium diskettes will last longer. This time usually translates to a much longer interval in actual use since the head is in contact with the media for only very short periods at a time. Eventually, however, a disk will wear out. The first part of a diskette to go is usually the directory cylinder, which is accessed anywhere from five to fifty times as much as the rest of the diskette.

These are a few of the most obvious and most common problems with disk drives. Such problems should be dealt with before attempting to use Super Utility 4 on a diskette in the same drive, lest worse errors occur.

What symptoms do you look for? Here is a list of some of the commoner ones, along with their possible causes. This list assumes that you are using good quality diskettes, and not diskettes bought at a garage sale. Poor quality diskettes can produce ALL of these symptoms and more!

1. FORMAT locks out sectors all over the disk: drive speed out of critical range, dirty heads, or worn pressure pad.

2. Intermittent or random errors during FORMAT: dirty heads, worn pressure pad.

3. Disk I/O errors on the high cylinders: head misalignment, worn pressure pad.

4. Inconsistent formatting (same disk will lock out sectors one time, then format perfectly five minutes later): dirty heads, worn media, worn pressure pad.

5. Disk I/O errors on a diskette from another drive: head misalignment.

6. Random I/O errors on directory cylinder: worn media.

7. Consistent I/O errors over the entire disk : drive speed outside critical range.

These symptoms should alert you to the possibility that your disk drive needs servicing, or that the diskette you are using is at the end of its "life" and should be replaced.

## CHAPTER 2 - ZAP UTILITIES

Bring up the main Super Utility 4 menu as described above, and press 1. You will then be presented with the ZAP Utilities menu.

```
__ Zap Utilities __

 1 _ Display Sectors          7 _ Reverse Sector Data
 2 _ Verify Sectors           8 _ Exchange Sectors
 3 _ Compare Sectors          9 _ String Search
 4 _ Copy Sectors            10 _ Sector Search
 5 _ Copy Sector Data        11 _ Read ID Marks
 6 _ Zero Sectors            12 _ Alter DATA Address Marks

Choice? #__
```

Select the routine you want by entering its number and pressing the <ENTER> key.

### I. DISPLAY SECTORS

When this option is selected, you will be presented with an additional prompt:

**Drive, Track, Sector ?**

Enter the number of the disk drive containing the disk you want to examine, then the track number and sector number you wish to display. Separate them from each other with a comma, for example,

**Drive, Track, Sector ? 1,0,1**

and press ENTER. If you wanted to read a diskette with a differing track count or density on the same drive, an alternative form would be 1T1S-40,0,1 where T1S indicates that the diskette is single density TRSDOS, and -40 means it is format-

ted for 40 tracks. See the previous chapter's discussion on configuring the system for more details. If you omit the track and sector, you will be shown the lowest sector on the lowest formatted track on the disk. If you enter D for track, Super Utility 4 will go to the configured directory location. If you enter an up-arrow for the track, then you will be shown the highest configured track on that diskette.

On this particular option only, you can also use two special symbols in front of the drive number. The first special symbol is a pound sign, or "#". This will cause Super Utility 4 to identify the diskette's density automatically, in case you are unsure of its density. This will work only with Display Sectors. This will also allow you to display non-standard sectors, however it will not scan the back side of a two-sided disk.

The second special symbol is an exclamation mark, or "!." This switches in automatic format recognition. Super Utility 4 will examine the disk and determine whether it was formatted with sector numbers from 0 to 17, or sector numbers from 1 to 18. This process takes a few seconds. This option is also available at any other routine which reads a diskette's directory. The main purpose of this routine is to permit Super Utility 4 to identify whether a disk is TRSDOS 1.3-formatted (sectors from 1 to 18) or TRSDOS 6-formatted (sectors from 0 to 17), and to render a disk readable to Super Utility 4's I/O routines. It is up to the user to decide whether Super Utility 4's identification of a disk's format as TRSDOS 1.3 or TRSDOS 6 is sufficient for his purposes or whether he should go in further.

The particular sector you requested will be displayed on your screen. It will look like what is shown on the top of the next page.

```
#    00#932F FDCB 03A6 300B FDCB 046E 2806 FDCB##/##.#0.##.n(.##
HEX  10#03E6 1E7B D3F2 7AD3 F3FD BE05 0618 2810#.#.{##z####...(.
DRV  20#FD72 0506 1CCD 8D0E C501 4000 CD00 05C1##r...##.#.@.#..#
 0   30#CDBB 0EF0 7E03 E603 B0D3 FD06 1210 FEAF###.#~.#.###...##
TRK  40#C97A D3F1 3A1B D0F6 4057 E610 0FCB 4920##z##:..#@W#..#I
 00  50#01AF B10E F318 E2CB 5001 8204 2010 FE0A#.##.#.##P.#. .#.
TRU  60#2806 CD89 0FD1 740E CD89 0FD1 6E0E FDCB#(.##..t.##..n.##
 00  70#037E 2803 3E0F C90E A2FE 0E38 060E A328#.~(.>.#.##.8..#(
SEC  80#020E F0CD 890F D9CC 0FE3 7E23 32C5 0F7E#..###..#.#~#2#.~
 13  90#2366 6F22 A30F E1C5 D5E5 CB61 CCF3 0ECD##fo"#.#####a##.#
STD  A0#800E CD00 D0FB DBF0 E67C E1D1 C1C8 CB57##.#..####¦#####W
DDD  B0#20E5 F5E6 1828 0BCB 67C5 C453 0EC1 F110# ###.(.#g##S.##.
     C0#D606 F147 3E00 CB08 D83C 18FA CD41 0F1E##.#G>.#.#<.##A..
     D0#760B F0A3 28FB EDA3 F3DB FD1F D03E C0D3#v###(######.#>##
     E0#E406 5D1D FE46 237A D3F4 DBF0 A328 F8ED##.P.#F#z#####(##
+00  F0#417A D3F4 EDA3 18FA 53AF D3E4 E1C9 0000#Az####.#S#####..
```

This display contains a wealth of information. The left-most column contains information about the display. **HEX** refers to the current modification mode base (see below). **DRV** and the number directly beneath it refer to the drive number just accessed. **TRK** and the number beneath it refer to the current track number, **TRU** and the number beneath it refer to the actual track number written on the disk, and **SEC** and the number beneath it is the sector being displayed on the screen.

Below the sector number is a three-letter code which identifies the particular data address mark written on the disk when the it was formatted. Super Utility 4 will identify two types of data address marks: **STD**, or standard, and **RPT** or "read-protected." The terms are those used by the manufacturer of the floppy disk controller chip and are not necessarily meaningful except insofar as they differentiate one type of address mark from another. All TRSDOS-type operating systems use a different type of data address mark for the directory track than for all the other tracks on a diskette. This is the major difference you need to be concerned about.

Below the data address mark identifier the density of the

diskette being examined will be displayed. This will normally be 0SD for single-density Model I diskettes and 0DD for double density diskettes. Super Utility 4 has the capability to recognize the density of a diskette it is reading and switch between single and double density if you specify the special "#" symbol in front of the drive number. The first character of the density ID string is "0" or "1" depending on which side of the disk is being read.

At the very bottom of the leftmost column of information you will see +00. This indicator is used by a special feature of Super Utility 4 called **decryption** which will be explained in a later section.

The next column gives you the relative byte numbers (in hexadecimal) of the data immediately to the right of the graphics border. "Relative byte" position simply means the position of a particular byte with respect to the first position, designated 00. Each row of the display shows 16 bytes with their values in hexadecimal format. Each group of four hexadecimal digits represents two bytes.

The data read from the disk sector is displayed between the two graphics borders, and to the right of each row is the ASCII representation of these bytes. Values which do not represent displayable ASCII characters (below ASCII 32) are displayed as periods. Also, if you are using a machine not equipped with lower case, any lower case alphabetic character will be shown in upper case; however the HEX value will always be accurate.

## I.1 Paging

You can use the arrow keys to page across sectors. The right-arrow key will page forward one sector, the left-arrow key will page back one sector. Pressing the up-arrow key will take you to the same sector on the next higher-numbered TRACK, while pressing the down-arrow key will take you to the same sector on the next lower-numbered track (unless you are

already at the lowest). In addition, there are a number of other paging controls, given in the table below.

TABLE 2-1 - PAGING CONTROLS

| KEY | ACTION |
|---|---|
| Right arrow | pages one sector higher (or to the lowest sector of the next track if the current one is the last for this track) |
| Left arrow | pages one sector lower (or to the highest sector of the preceding track if the current one is the lowest for this track) |
| Up arrow | pages one track higher, same sector |
| Down arrow | pages one track lower, same sector |
| SHIFT-Right arrow | pages one sector higher but will not leave current track (see note) |
| SHIFT-left arrow | pages one sector lower but will not leave current track (see note) |
| SHIFT- ) | displays highest sector on current track |
| SHIFT- ( | displays lowest sector on current track |

| | |
|---|---|
| **SHIFT-Up arrow** | pages to the same sector on the highest track (defined in the configuration tables) |
| **SHIFT-Down arrow** | pages to the same sector on the lowest track |
| **R** | displays Track 0, Sector 0 (Sector 1 on TRSDOS 1.3 and 2.7DD. |
| **T** | displays prompt for new Track, Sector |
| **S** | displays prompt for new sector |
| **CLEAR** | displays prompt for new Drive, Track, Sector |
| **. or >** | pages to the next higher VALID sector |
| **, or <** | pages to the next lower VALID sector |
| **number keys 0-9** | displays the correspondingly numbered sector on the current track |
| **BREAK** | returns you to the ZAP Utilities menu |
| **SHIFT-BREAK** | returns you to the main menu |
| **@** | enables DECRYPT mode (see below) |
| **H,D,B,O,Q,A** | Sets modification mode BASE to hexadecimal, decimal, binary, octal (O and Q are equivalent) or ASCII. |
| **M** | Enters modification mode. |

**Note 1:** Shift-right arrow and shift-left arrow may occasionally display sector numbers that are beyond the expected range. This will happen when Super Utility 4 tries to read non-standard sectors and is not a cause for concern.

**Note 2:** Depending on the case setting, you may need to use either @ or shift-@ to enter decryption mode.

**Note 3:** If you are attempting to display sectors with non-standard sector numbers, use > and < instead of the arrow keys.

As you can see, the ZAP utility provides you with tremendous flexibility in searching through a disk. You can view any sector on the disk with relative ease.

One of the most powerful commands in Super Utility 4 is the L or LAST command. When you are viewing a sector and exit to the menu, either deliberately or accidentally, you can always return to the sector you were viewing by calling up the drive, track and sector prompt line and simply entering d L where "d" is the drive number and "L" takes the place of the track and sector values (the space is mandatory). You will immediately be returned to the last sector you were viewing.

The value of LAST is updated by a number of routines. When performing a sector comparison, the last sector in which a mismatch was found will update LAST. Thus if you were comparing two disks and wanted to scan the mismatch, you would merely press CLEAR to stop the routine, request DISPLAY DISK SECTORS, and when the prompt for drive track and sector comes up, enter the drive number followed by L and you will be taken to the sector where the mismatch occurred.

When performing a string search (see below) the last sector in which a string match was found will also update LAST.

## I.2 Modifying the contents of a disk sector

The ZAP utility gives you the powerful ability to directly modify the contents of a disk sector. This should only be undertaken with great care, since careless modification can make a file or a disk totally useless for other purposes.

Display the sector you wish to modify using the instructions given above. Then look at the leftmost column on your screen. On the second row you will see **HEX** displayed. Super Utility 4's ZAP utility gives you the ability to modify the contents of a disk sector in either Hexadecimal, Decimal, Octal, Binary, or ASCII form. The default is hexadecimal, and this is what the HEX means. Press D on your keyboard, and you will see it change to DEC. Now press A, and ASC will be displayed, meaning that ASCII modification is enabled. The

keys for switching the modification case are H for hexadecimal, O or Q for octal, D for decimal, B for binary, and A for ASCII.

Press H again. Now press M. You will see a pair of flashing cursors appear in the data portion of the display, one in the hexadecimal portion, and another in the corresponding position in the ASCII display to the right.

The cursor may be positioned anywhere in the data portion using the arrow keys. The right arrow moves the cursor one byte to the right, the left arrow moves it one byte to the left. The up and down arrows move the cursor up and down the rows. By holding down the SHIFT key and pressing one of the arrow keys, you can position to the far ends of the display. For example, pressing SHIFT and the right arrow key will position the cursor to the rightmost byte on that row. Pressing SHIFT and the down-arrow key will move the cursors to the bottom row of the display. The current position of the cursor, in relative byte format, will be displayed at the upper left of the screen. Pressing the CLEAR key will return the cursor to the upper left position (byte 00). The other positioning keys available in MODIFICATION MODE are given in the following table.

**TABLE 2-2 - Modification Controls**

| Key | Action |
|---|---|
| **H,D,B,O,Q,A** | selects modification mode base (Hexadecimal, Decimal, Binary, Octal or ASCII). This should be done before entering Modification mode. |
| **SHIFT-ENTER** | reset modification mode (permits re-selection of H, D, O, Q, B, or A). |
| **Right Arrow** | moves cursor one byte to the right. |
| **Left Arrow** | moves cursor one byte to the left. |
| **Up Arrow** | moves cursor up one row. |

| | |
|---|---|
| Down Arrow | moves cursor down one row. |
| SHIFT-Right Arrow | moves cursor to last byte on row. |
| SHIFT-Left Arrow | moves cursor to leftmost byte on row. |
| SHIFT-Up Arrow | moves cursor to top row. |
| SHIFT-Down Arrow | moves cursor to bottom row. |
| CLEAR or S | returns cursor to relative byte 00 (leftmost byte of top row). |
| Q | moves cursor to the last byte of the bottom row. |
| G + relative byte no. | moves cursor to the specified relative byte location (not active if in ASCII modification mode). |
| L + numeric input | moves cursor to the next occurrence of specified numeric input, i.e., "L3F" will place cursor on the next occurrence of 3F in that sector (not active in ASCII modification mode). The numeric input MUST be in the current modification base. |
| + and numeric input | moves the cursor that many bytes forward from its current position. |
| - and numeric input | moves the cursor back that many bytes from its current position. |
| > | insert data at current cursor position and move the rest of the data one byte to the right. |
| < | delete the data beneath the cursor and shift the rest of the data one byte to the left. |
| P + numeric input | copy the byte beneath the cursor the specified number of times to the right. The numeric input must be in the current modification base. |
| Z | zeroes out display and holding buffer from the current cursor position on. |
| ENTER | terminate modification mode. |
| BREAK | abort modification mode, return to ZAP utilities menu. |

**SHIFT-BREAK (F1)** abort modification mode, return to main Super Utility 4 menu.

Note that the keys being used in modification mode are the same as those used in PAGING mode, but have entirely different actions.

Extreme care should be taken when using the modification mode controls, as the result may not be what you expect.

By typing a valid key (that is, a key valid for the current modification mode you are in), you will enter that value into the display at the current cursor location. For example,

```
   00  FE11 3ED0 D3F0 2102 0022 EA43 _F32
HEX 10  CD3E 43FE 0128 0CFE 0220 E7C0 3E43
```

In this case the cursor is under F0 on the first row. If you now enter a valid numeric digit, say 7, the first row would look like this:

```
   00  FE11 3ED0 D3_  2102 0022 EA43 AF32
```

Note that the cursor has changed, and the byte you are modifying has temporarily disappeared. Super Utility 4 is reminding you that you need to enter another hex digit to complete the modification of that byte. Now type D. The row now looks like this:

```
   00  FE11 3ED0 D37D 2102 0022 EA43 AF32
```

F0 has been replaced with 7D, and the cursor has moved one byte over.

The cursor will remain in its changed state until you have entered the necessary number of digits to enter a new value. This will vary depending on the modification base you are using. For example, if you are in BINARY modification, you

will normally have to type in 8 binary digits before the new value appears on the display. If you wish to terminate the input early before entering all the digits, simply press ENTER. The binary value will be padded on the left with the requisite number of zeroes so as to evaluate properly. To abort an entry simply type in an invalid digit and press ENTER.

No matter what modification base you use, the display will always appear in hexadecimal, with the corresponding ASCII equivalents on the right.

In ASCII modification, all keys on the keyboard except for the arrows, the greater-than and less-than symbols (> and <), BREAK, CLEAR and ENTER are valid for input. On the Model 4, the CTRL key may be used for ASCII modification in combination with other keys; however you should be aware that certain control-key combinations may be seen as commands by Super Utility 4, and will be acted upon as such. For example, a CTRL-D will terminate the modification mode immediately. This is not a bug, and should not be considered as such.

Pressing ENTER terminates modification mode. You will now be presented with the prompt:

**U>pdate, R>eturn to modify, or C>ancel ?**

The modified sector will be written out to disk when you press ENTER or explicitly choose the Update option (type U and press ENTER). The sector data will be written back out to disk, and then re-displayed. Remember, UPDATE is the default, and hitting ENTER alone will select it.

If you enter R, you will be returned to the sector display, and the disk will not be updated. You can continue making modifications. The display will contain the modifications you have made up to that point.

Pressing C cancels the modification session. All your changes will be canceled, and the sector will be re-displayed with its original data intact.

### I.2.1 Using the special modification controls

The greater-than key (>) permits you to insert data into the display without having to retype what is already there. All the other bytes will be shifted one byte to the right, with the last one being lost. Conversely, the less-than key (<) will delete the byte beneath the cursor and shift all other bytes one place to the left to fill in the gap. The last byte, which was shifted OUT when you pressed ">", will NOT be brought back in but will be replaced with 00.

While in a numeric (not ASCII) modification mode, you can use a number of other keys to make changes to the displayed sector data with ease.

The G key allows you to move directly to any position in the displayed sector without having to use the arrow keys. Simply type G followed by the relative byte position you want to go to, and the cursor will be placed on that position. The number following G will be interpreted in the current modification base (except when the modification base is ASCII, of course). For example,

```
   00  00FE 11CD C901 FD21 0000 3A02 4257 1EE4
HEX10  0651 CDAB 4220 5801 E650 FE50 2056 2A16
                      .
                      .
                      .
   60  1823 10F6 18CA CD7E 426F CD7E 4267 C921
```

The cursor is positioned at relative byte 5, which holds the byte 01. We want to position the cursor to relative byte 67 hex, which is 7E. To do this, just type G67. The cursor has now been positioned where we want it:

```
   00  00FE 11CD C901 FD21 0000 3A02 4257 1EE4
HEX 10  0651 CDAB 4220 5801 E650 FE50 2056 2A16
                        .
                        .
                        .
   60  1823 10F6 18CA CD7E 426F CD7E 4267 C921
```

The LAST variable is also updated by this command, so that if you reposition the cursor and later wish to return to the original byte, simply typing GL will get you there. The use of the L will allow you to reposition the cursor with a minimum of keystrokes.

The + and - keys operate in a similar fashion. Typing a + followed by a number in the current modification base will move the cursor forward that many bytes from its current position. If the number entered was such that it would cause the cursor to leave the display, the cursor will be positioned at the last byte of the display. Typing a - followed by a number will move the cursor back that many bytes from its current position.

Another handy control is L. This permits you to locate a particular byte on the display and position the cursor over it. In the example above, suppose the cursor was at relative byte 00, and you wanted to locate the byte whose value was 21 hex. Type L21. The cursor will immediately be positioned over relative byte 7. Now press L21 again. The cursor is now positioned at relative byte 6F hex, which is also 21H! Remember however, that like the G command, the number which follows L is also interpreted in the current modification base.

Instead of repeatedly pressing L21, however, you could also enter LL. The second L refers to the "last value entered" and would produce the same results. This is also available in the G command.

For quick positioning to the start and end of the display, the S and Q keys are available. The S key will move the cursor back to the first byte on the display, while the Q will move it to the last byte on the display.

Sometimes you want to duplicate a certain byte a specific number of times. You can do this very easily with the P key. In the example above, suppose you wanted to replace the entire first row with 00. Position the cursor over the first byte, which is already 00, press P, and 0F. The rest of the row, 15 bytes all in all, have been replaced with 00 (0F hex =15 decimal). As with the G and L commands, if your modification base was decimal, you would have had to enter P15, since the number would be taken in the current base. Note that the command P00 (P zero zero), which would normally be interpreted as "propagate this byte 65,536 times", is invalid and will not be accepted by Super Utility 4.

The S, Q, G, L, +, - and P commands are obviously not usable when you are in ASCII modify mode, because in this mode all keys except for the four arrow keys, >, <, BREAK and ENTER are valid input.

For zeroing out the display and the contents of the holding buffer, you may press Z. This will immediately remove all data from the holding buffer and replace them with 00 bytes starting at the current cursor position.

## I.3 Bit-shift operations

While you are in paging mode, you may perform various bit-shift and bit-rotate operations on the displayed data. These operations are analogous to those performed by the Z-80 assembly language instructions RLCA, RRCA, SLA, and SRL. They involve shifting the bits of an 8-bit byte a specified number of times to the left or right to form a new value. In shift operations, the bits at the far end are usually lost, while in

rotate instructions, the bits at the far end are "rotated" into the opposite end. For example, take the binary representation of the value 85 (decimal):

0 1 0 1 0 1 0 1

To shift this byte 1 place to the right would yield the value

0 0 1 0 1 0 1 0

or 42 decimal. The trailing 1 bit has been lost. Shifting the original number left one place would give you

1 0 1 0 1 0 1 0

or 180 decimal. Shifting it left two places yields the binary value

0 1 0 1 0 1 0 0

which is 94 decimal.

Rotate operations involve moving the bit that would normally be lost at the far end into the opposite end of the bit string. Using our example, 0 1 0 1 0 1 0 1, If we rotate this value right one bit, the result would be

1 0 1 0 1 0 1 0

or 180 decimal. Rotating it left one bit yields

1 0 1 0 1 0 1 0

also 180 decimal! Rotating it left two bits gives us

0 1 0 1 0 1 0 1

bringing us right back where we started, at 85 decimal. Other numbers would give other results, of course.

Shift and rotate instructions can be carried out for any number of places you wish, up to 7 (the width of one byte). In Super Utility 4, if you wish to execute a shift or rotate instruction, you must first be in the paging (not modification) mode. The shift or rotate operations are carried out on ALL the displayed data bytes simultaneously. To see how this works, first display a sector and make sure you are in the paging mode. Now press @. Toward the bottom of the display at the left side, you will see the prompt

DCR

(DCR stands for "decryption" -- this routine can be used to investigate sectors which may have been encrypted either by bit shift, logical or increment/decrement operations).

To execute a Rotate Right operation, press RR followed by the number of places you want to rotate each byte on the display. You will see the bytes change as soon as you hit ENTER. To execute a Rotate Left operation, type RL followed by the number of places to rotate and press ENTER. SR followed by a number executes a shift right operation, and SL executes a shift left operation.

In addition, this particular routine is capable of carrying out a variety of logical operations on the displayed data. The commands are listed on the top of the next page. Each command must be followed by a numeric input (the base must be properly identified by appending H, B, O, or Q -- decimal is the default -- to the number) and ENTER:

| | |
|---|---|
| A | AND the displayed bytes with given input |
| O | OR the displayed bytes with given input |
| X | XOR displayed bytes with given input |
| + | ADD the given input to each byte and display the result, modulo 256 |
| - | SUBTRACT the given input from each byte and display the result modulo 256 |

You can also command the computer to increment or decrement the display automatically any given number of times. This may be useful if you wish to see if any ASCII words in the displayed sector have been encrypted by adding or subtracting a constant value to its ASCII value. It may be necessary to type "@" again to re-enter decryption mode as some of the commands automatically take you back to normal paging mode.

While executing this operation, you can further tell the computer whether to update the ASCII side of the display only (the default), by entering "*" as the first character in response to the DCR prompt, or both the HEX and ASCII sides by entering ":".

You can increment or decrement the display one bit at a time in decryption mode also. You would enter decryption mode as usual, and select whether you want the HEX side only modified, the ASCII side only modified, or both. Then you would enter an up-arrow if you wanted an increment operation, or a down-arrow if you wanted a decrement operation (you cannot do both simultaneously). Finally you can control the speed at which the computer updates the display.

Suppose you wanted to increment the ASCII side of the display and view the results. In response to the DCR prompt, you would enter * followed by an up-arrow (for incrementing) and finally a decimal number from 1 to 255 for speed (1 is the fastest, 255 is the slowest). As soon as you press ENTER you will see the ASCII side of the display begin to change.

You may stop this automatic operation anytime by pressing @, or you can pause it momentarily by pressing the spacebar. The display will stop updating when all the bytes reach the maximum value, FFH or 255 decimal. Entering a "!" makes the changes permanent and must be the first character of a new DCR prompt line.

This operation can be used to encrypt your own messages very easily. Suppose you had a sector of code which, among other things, contained the ASCII string "MYPROGRAM" and you wanted to hide the string by XOR'ing each byte with 88H. Enter decryption mode and make note of where the string MYPROGRAM starts in the sector. Then XOR the entire sector with 88H and use the "!" command to make the change permanent in the buffer. Now enter modification mode and place the cursor at the location where the string starts, and type in the string MYPROGRAM normally. Exit modification mode and go into decryption mode again. Now once again, XOR the entire sector with 88H. The rest of the sector will now be back to normal but your string "MYPROGRAM" will now become encrypted! Finally, use the "!" symbol to make the changes permanent and write the sector out to disk.

The +00 on the bottom left of the display is updated to show the amount the data bits are shifted, rotated, logically operated on, or incremented or decremented by a DCR operation. The symbol preceding the two numbers will change to reflect the current operation, in accordance with the table above. Pressing ENTER in response to the DCR prompt will reset this number to 00.

## I.4 Error Recovery

Should any I/O errors occur while Super Utility 4 is attempting to read or write to a disk, a message describing the type of error encountered will be displayed, for example,

**Sector NOT FOUND!**

or

DATA CRC ERROR!

and you will be given an option to retry the I/O operation with the prompt,

**R>etry, S>kip, C>ontinuous, N>onstop or Q>uit ?**

for as many times as necessary. You may press **R** (or just ENTER) to retry the I/O operation once. If the error occurred as a result of some momentary condition, this is usually sufficient to correct the situation. However, if the error continues to appear, you may want to type **C** for continuous retries. This will force Super Utility 4 to retry the I/O operation until it gets it right, or CLEAR is pressed. Pressing CLEAR will restore the retry prompt. Also, break and shift-BREAK (F1) may be pressed from the keyboard to abort the operation and return you back to a Super Utility 4 menu. Pressing **N** will produce the same results as Continuous, but in addition will automatically place Super Utility in continuous mode on subsequent errors.

Pressing **S** will immediately bring up the sector display. Depending on the error encountered, this may or may not contain any data. If the sector could not be read at all, the display will contain all 00 bytes (the buffer is zeroed out before a sector read is performed). If Super Utility 4 was successful in partially reading the display, some data will be present in the display. However, you should not assume that what is shown on the display is an accurate representation of the data. If the error was simply a data CRC error, then the data may be intact. However, some bytes of data could also be garbled. You should examine the display closely and come to your own determination of whether the data displayed is intact or at least repairable. If you decide the data is intact, or have carried out the necessary repairs, then writing the sector back out to disk will automatically correct any CRC

errors which may exist as long as the media is physically intact.

Pressing Q will abort the operation in progress and return you immediately to the ZAP Utilities menu.

## II. VERIFY SECTORS

This option allows you to scan all or part of a disk for conditions which would produce I/O errors. This routine does not check the data, but rather whether or not sectors are readable. You will be prompted for the drive, track and sector to verify, and the number of sectors to verify. The program will then proceed to read these sectors and report any errors which it encounters.

When errors are encountered, the program will display an error message followed by the prompt to R>etry, S>kip C>ontinuous, N>onstop, or or Q>uit. If you press R, the program will attempt to read that sector again. If it succeeds, it will continue. If it does not succeed, it will redisplay the error message.

Pressing C or N for continuous re-try will force the program to re-read the bad sector UNTIL it gets it right. If it cannot read the bad sector at all, you may exit by pressing CLEAR, BREAK, or SHIFT-BREAK depending on where you wanted to go next.

Skip tells the program to skip the bad sector and continue. The program will keep an internal count of the bad sectors encountered and will report the total number of bad sectors at the end of the operation.

Quit simply brings the verification routine to an end at that point. You will be brought back to the ZAP utilities menu.

## III. COMPARE SECTORS

This utility permits you to compare the contents of two different sectors and is useful if you want to verify that a backup operation made an exact copy. This routine will also check for data address mark mismatches.

When this option is selected, you will be prompted for the drive, track and starting sector number of the source disk. The program will then ask you for the number of sectors to compare (default is to the end of disk). When you have entered this number, you will then be prompted for the drive, track and starting sector location on the destination disk for the comparison operation.

After this, you will be asked, "Prompt for disk mounts?" If you answer Y (yes), you will be told when to swap disks. This is especially useful when you have only one drive in which to do the comparison. The number of disk swaps will be determined by the amount of memory available to Super Utility 4 for buffer space. On a two-drive comparison, of course, no disk swaps are necessary. However, you can use the disk mount prompts on a two drive system to check the readability of a disk on two different drives, should you suspect that the drive hardware is at fault.

## IV. COPY SECTORS

This option will allow you to copy sectors from one disk to another, or from one location on the disk to another. Only full sectors are copied. When this option is selected, you will be prompted for the drive, track and starting sector of the source disk, and the number of sectors to be copied. Then you will be asked for the drive, track and starting sector number of the destination disk. As with the COMPARE SECTORS option, you will then be asked if you want to be prompted for disk mounts. If you reply Yes, you will be told when to swap

disks. If your source and destination drives are the same, the disks you swap must be the same disk type.

The data will then be copied over to the new locations, leaving the original locations intact. The track and sector id fields will not be copied, but the correct DAMs and the actual contents of the sectors will be transferred.

This procedure will NOT work correctly if used on sectors with non-standard lengths, i.e., lengths other than 256 bytes. Attempting to copy such sectors may have disastrous results and should not be attempted.

Note that this operation is not reflected in the diskette's directory. You may use this routine to copy a file from one location to the other, but your directory will not show the file in its new location.

## V. COPY SECTOR DATA

This option will allow you to copy partial sector data onto a new sector. You will be prompted for the drive, track and sector of the source disk, the relative byte number within that sector where the copy is to begin, and the number of bytes you wish copied. Then you will be prompted for the drive, track and sector of the destination disk, along with the starting byte position for the copy, and whether or not you want disk mount prompts. The copy will then proceed. You will be advised upon completion of the routine whether any disk I/O errors occurred during the copy.

This routine will allow you to copy from 1 to 65535 bytes of information to a new location of your choice. The same restrictions as COPY SECTORS apply to the use of this routine on non-standard length sectors.

## VI. ZERO SECTORS

This option will totally remove the data from the specified sectors, setting the entire contents of the sectors to 00 and resetting the data address marks to STD (RPT on Model 4 TRSDOS). You will be prompted for the drive, track and starting sector number for the operation, along with the number of sectors to zero. **BE CAREFUL WITH THIS ROUTINE!** Upon pressing the ENTER key the operation will immediately be carried out, and there is ABSOLUTELY NO CHANCE of recovering the data once a sector has been zeroed out!

Do not use this routine to zero out directory sectors, as the directory sectors will be written out with the wrong DAMs. Use the "Zero Unused Entries" routine in the PURGE UTILITIES menu.

## VII. REVERSE SECTOR DATA

This routine simply takes the data of a specified sector and reverses it, so that the byte that as in relative position 00 is now in relative position FFH and so on. This routine may be useful in creating disk protection schemes for machine language programs and data. If multiple sectors are specified then each sector will be worked on independently. That is, the reversal of sector data takes place entirely within each individual sector.

## VIII. EXCHANGE SECTORS

This routine will exchange the data contained in one or more sectors with the data in another set of sectors. You will be prompted for the source drive, track and starting sector number, along with the number of sectors to exchange. Then you will be prompted for the destination drive, track and starting sector number and whether or not you want to be prompted for disk mounts. Upon pressing the ENTER key the data in each

sector on the source disk will be exchanged with the corresponding sector on the destination disk. Upon completion of the operation you will be advised of any disk I/O errors that may have occurred.

## IX. STRING SEARCH

Up to an entire disk may be searched for a given ASCII string, BYTE list or WORD list (a word equals two bytes, a total of 16 bits) using this routine. Additionally you may optionally specify a replacement string which the routine will insert in place of the target string whenever the target string is found.

You will be prompted for the drive, track and starting sector for the search, and the number of sectors to search. Next, you will be prompted to enter the search string, and lastly, the replacement string (if you do not wish to perform any replacement, merely hit ENTER in response to this prompt). If the length of the replacement string is shorter than that of the search string, only that portion of the search string which corresponds to the length of the replacement string will be replaced. This will permit you, for example, to search for all occurrences of "John Brown" and change them to "Carl Brown" by simply giving "Carl" for a replacement string.

If the replacement string is longer than the search string, it will be truncated to the length of the search string. The disk will then be searched, and upon completion, the location of each match and the total number of matches found will be displayed.

The search string must be contained wholly within a sector; that is, part of it cannot reside in one sector with the rest in the next sector. If this is the case, the routine will not find it. If you suspect that this may be the case, try performing a search on only a portion of the string.

To search for an ASCII string, simply enter it when prompted. For example, if you wanted to search for an occurrence of the name "Jim", you would enter,

"Jim"

in response to the prompt. Enclosing the word in double quotes means that you want the search to match upper and lower case EXACTLY. If you wanted to find occurrences of "Jim" and "JIM" you would enclose the search string in single quotes. Single quotes tells Super Utility 4 to search for the string in a case-independent fashion, that is, upper and lower case differences will not matter.

If you wanted to find all occurrences of "Jim" and "Tim" on the disk, you could enter,

"?im"

in response to the prompt. The "?" is a wild-card character, meaning, in effect, "I don't care what's in this position, match it with whatever's there and show it to me anyway."

To search out BYTE values, when you are prompted for the search string, enter a series of values in the range 0 to 255 in any valid numeric base separated by commas or spaces. For example,

10H,13H,CDH

which in decimal would be

16,19,205

Note that each element of the reply is separated from the rest by a comma (they may also be separated by spaces).

The word search is slightly different. This is normally used to locate sixteen-bit address references in a machine

language program. The Z-80 stores such addresses in reverse order, that is, the least significant byte first, followed by the most significant byte. Thus, the address reference 7F42H would be stored in a machine language program as 42 7F.

The word search routine assumes this to be the case, so that when you enter a two-byte value, it will automatically reverse them before starting the search. Thus, if you wanted to find the two bytes 7F 42 in that exact order, you must enter them as 427FH or, alternatively, use the byte search mode to look for the byte pair 7FH,42H. Note, however, that this will not work when you are looking for byte pairs that end in 00, for example 3300H. The reason for this is that when Super Utility 4 evaluates your ASCII input into binary, it will skip over leading zeroes. Thus if you were to enter 0033H, it would be evaluated as 33H. For these cases you must use a BYTE search.

The word search is specified by entering 16-bit values on the prompt line in any valid numeric base, although due to the reversal which takes place it is probably easier to use hexadecimal (easier for YOU, that is). The values should be separated from each other by commas or spaces. For example,

**7F42H,CD30H,402DH**

Notice how the bytes are grouped in pairs and each pair is separated by commas. This will result in a search for the bytes (in hexadecimal) 42 7F 30 CD 2D 40 in that order.

If you wish to specify a replacement string, you must make sure that the replacement string is the same length as or shorter than the search string, or the replacement string will be altered (truncated to the right length). You could specify an ASCII search string and then replace it with a BYTE or word string, but it is usually easier to enter the search and replacement strings in the same format.

However, it is also possible to mix different modes in a single search or replacement string. That is, ASCII strings, byte, and words can all be placed on the same line. For example,

"Test",42H,79E0H,'disk'

is a perfectly valid string search specification.

Finally, note that if "@" is the first character of a search string, then it will be assumed that the target string has been encrypted according to the current setting of the decryption routine. The search string will be encrypted simirly before the search is conducted. If a byte string is given along with "@" then the byte string will be created first, then the string encrypted.

## X. SECTOR SEARCH

This option is useful in searching out duplicate sectors on one or more disks. You will first be prompted for the drive, track and sector number of the sector that is to form the search template. Then you will be asked for the drive, track, and starting sector number for the search, followed by the number of sectors to search. When the routine completes, it will display the total number of matches that it found, and their locations on the target disk.

## XI. READ ID ADDRESS MARKS

This routine will examine a disk and identify the track and sector data written on the target disk. This will identify any false or non-standard sectors on the disk and help you determine just how a disk was formatted. You will be prompted for the target drive to examine, and the program will start reading track 0 and present the information on the display. The information will scroll by very rapidly, but can be

stopped at any time by pressing the spacebar and resumed by pressing ENTER. It may also be exited by pressing the BREAK key. It will advance up one track whenever you press the UP-ARROW and down one track when you press the DOWN-ARROW. Additionally, you may proceed directly to the highest formatted track on the disk by pressing SHIFT-Up arrow, or to the lowest formatted track on the disk by pressing SHIFT-Down arrow.

You may also press S to force the routine to read only single density sectors, or D to make it read only double-density sectors. In addition, when dealing with double-sided diskettes, pressing F or ' (apostrophe) will force a read of the front side, while pressing B or " (double quote) will cause the back side of the disk to be scanned. These keys are active while the display is scrolling.

If the routine encounters an unformatted track, it will display the error message "ID ADDRESS READ ERROR" or "TRACK NOT FORMATTED."

Sometimes this routine will return an error message for a sector, then when that sector is scanned again, there will be no error message. You should keep in mind that this routine is reporting faithfully what the floppy disk controller chip is sending, and on occasion the FDC will "miss" a sector and report back an error condition. On a normal operating system, this would immediately cause a retry. Super Utility 4, however, is designed to catch errors in media and formatting, and so it does not perform any retries at all. It will immediately report the error returned by the disk controller.

Normally such random errors in the Address Marks scan is not a cause for concern. If a sector comes back bad consistently rather than randomly, however, then one can assume that there is a real error on that spot of the disk.

Initially, the routine will display seven columns of information, as follows:

```
. . . . . .
. ** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **  .
.   Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved  .
. Source  Track Head Sector Length CRC1  CRC2  ckCRC   Data  .
. :0D'= 02  02    00   06     01    8EH   F3H                .
. :0D'= 02  02    00   07     01    BDH   C2H                .
. :0D'= 02  02    00   08     01    ADH   FCH                .
. :0D'= 02  02    00   09     01    9EH   CDH                .
. :0D'= 02  02    00   10     01    CBH   9EH                .
. :0D'= 02  02    00   11     01    F8H   AFH                .
. :0D'= 02  02    00   06     01    8EH   F3H   YY     STD   .
. :0D'= 02  02    00   13     01    52H   09H   YY     STD   .
. :0D'= 02  02    00   03     01    71H   06H   YY     STD   .
. :0D'= 02  02    00   10     01    CBH   9EH   YY     STD   .
. :0D'= 02  02    00   17     01    14H   17H   YY     STD   .
. . . . . .
```

The leftmost column is labeled SOURCE. This gives you information as to which drive you are reading, and the density of the disk in that drive. For example, :1D= 01 indicates that the disk in drive 1 is double density, and the read/write head is positioned over physical track 1.

The next column is labeled TRACK, and this is the track number which is actually recordEd on the disk. Some protected disks use non-standard track numbers, and if the disk you are reading is one of these, the track column would not necesrily agree with the actual track number to its left.

The third column, HEAD, is the head number actually recorded on the disk. In double-sided disk drives, the head number could indicate which side of the disk is being read.

The fourth column is SECTOR. These are the sector numbers as recorded on the disk, and will not appear in any special order. Sectors which are physically located side by side on a disk will not be numbered consecutively. Also, some protected disks will assign false sector numbers to prevent them from being read by standard operating systems.

The next column, labeled LENGTH, is a coded indication of the amount of data contained in the sector. In the IBM convention which TRSDOS and most TRSDOS-compatible systems use, a length of 00 means 128 bytes per sector, and a length of 01 means 256 bytes per sector. If you are scanning a standard disk, this should display 01. In the Model 4 and double-density Model I disks, the only acceptable values for LENGTH are 00 through 03, corresponding to lengths of 128, 256, 512, and 1024 bytes respectively.

If the disk is not formatted with the IBM conventions, then a 00 length byte indicates 1024 bytes per sector, and a non-zero length byte multiplied by 16 will yield the actual number of bytes in the sector. By this convention, a non-IBM length byte of 10H (16 decimal) would be equivalent to an IBM length byte of 01. Non-IBM conventions are possible on Model I single density disks only, as the double-density controller used by the Model 4 does not use the IBM/NBM indicator.

Note that the length code indicates the number of bytes as recorded on the disk's data fields; it does not necessarily mean that there are that many bytes actually present in the sector (again, this may be used by disk protection schemes to lay a false trail).

The next two columns are labeled CRC1 and CRC2. CRC stands for "Cyclic Redundancy Check," and the two bytes are the result of a lightning-fast calculation made by the floppy disk controller chip on the sector data. When a sector is read back in by the computer, the disk controller chip recalculates the CRC's and compares them with those recorded on the disk. If the bytes recorded on the disk fail to match those

recalculated by the FDC, it signals an error condition, which normally appears on your DOS display as "CRC error" or "Parity error during read."

Planting false CRC bytes is another favorite sport of disk protection experts. They will override the FDC's calculation and insert their own bytes in place of the true CRC's, thus forcing standard operating systems to ALWAYS signal an error when attempts are made to read their disks.

If you press the X key while the display is scrolling, you will produce an additional three columns of information on the screen. The first, immediately to the right of the CRC2 column, is labeled CKCRC. This is the result of Super Utility 4's reading of the sector, which forces the FDC to report back the CRC bytes for the sector. The result is indicated by two letters. The letter to the left pertains to the ID field's CRC, and the letter to the right pertains to the data field CRC. A "Y" indicates that that CRC byte on the disk is correct, and an "N" means that there is a discrepancy between the recorded CRC byte and the actual byte arrived at by the FDC on the basis of the sector data. If the ID field CRC is bad, then the data field CRC check will appear as "*" since it is impossible to check the data.

Finally, the DATA column tells you what data address mark was used in formatting. This will display STD for standard, DDT for deleted data, RPT for "read protected," and UDF for user-defined. As we said before, the names do not really mean anything and are used simply to differentiate one type of address mark from another.

The Model I floppy disk controller chip is capable of reading and writing all four types of data address marks, or DAM's. However the double-density controller of the Model 4/4P can only produce and recognize two types of data address marks: STD and RPT, even if it is reading or writing a single-density disk.

NOTE: when X is pressed to bring up the additional three columns of display, the keyboard will tend to become sluggish as it will not be scanned as often as when the normal display is scrolling past. Thus if you wish to use the arrow keys to move to another track, for example, it is necessary to hold these keys down for a couple of seconds before the desired effect is achieved.

## XII. ALTER DATA ADDRESS MARKS

This routine will allow you to alter the data address marks on a diskette to something other than the standard marks used by your disk operating systems, and may be a good way to produce your own "protected" diskettes. You will be prompted for the drive, track and starting sector for the alteration, and the number of sectors to alter. Finally, the program will prompt you for the type of data address mark you wish to use. Answer S for standard, or R for read-protected.

Again, care should be taken when using this option, as you can wind up with a disk that no operating system can read correctly.

# CHAPTER 3 - PURGE UTILITIES

Bring up the main menu and press 2. This will display the Purge Utilities menu:

```
** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **
  Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved

__ Purge Utilities __

 1 _ Kill Selected Files        6 _ Zero Unused Entries
 2 _ Kill by Category           7 _ Zero Unused Grans
 3 _ Remove System Files        8 _ Change Disk Name
 4 _ Remove Passwords           9 _ Change File Params
 5 _ Disk Directory

Choice? #__
```

The Purge Utilities operate mainly on the directory of your system or data diskettes. They provide a very fast and very convenient way of killing (and recovering!) files from a disk, and even include facilities for checking your disk directories for errors, and removing all traces of a file from a disk by zeroing out the sectors which it occupied. This last is very useful in maintaining security.

## I. KILL SELECTED FILES

The first choice on the purge menu is KILL SELECTED FILES. This will permit you to scan the directory of a diskette and "tag" certain files for removal. The files are shown on the screen, along with a cursor. Active files are shown surrounded by left and right square brackets and the non-active file entries (killed files) are shown surrounded by graphics blocks.

Note that the KILL command of TRSDOS 1.3 (Model III) and TRSDOS 2.7DD (Model I) remove all traces of a file from the directory, so that you may not see any non-active file entries on the display when dealing with such disks. Super Utility 4's purge routines do NOT remove all traces of a killed file from the directory in order to leave open the possibility of recovering killed files if needed. Similarly, TRSDOS 6.x and other non-RS systems merely tag removed files as inactive rather than wiping all traces of them completely from a diskette directory.

You may position the cursor at any filename by using the arrow keys. By pressing K while the cursor is positioned at a particular file, you "tag" that file for killing. You can hit BREAK at any time to abort this operation without doing any damage, since the purge operation does not actually take place until you give the command to write the updated directory back onto the disk.

If you press C instead of K, you will not only kill the selected file, but also physically remove all traces of its entry from the directory. Pressing the CLEAR key at any time will zero out ALL unused directory entries, including slots containing inactive file entries.

Pressing N will advance the cursor to the next filename in the list and is analogous to using the right-arrow to position the cursor.

By pressing **R** while the cursor is positioned at a non-active file entry, you can restore that file to the directory as an active file. However, you should exercise caution when recovering killed files, as other files may have overwritten parts of it. There is no guarantee that a recovered file is still intact, unless you recover it before any other write operations to the disk have taken place.

Pressing the **CLEAR** key will remove all inactive files from the display and zero out their entries in the directory which is buffered in memory. When CLEAR has been pressed, R will no longer have any effect. When the directory is written out to disk, the entries of all inactive files will have been zeroed out, making future recovery impossible, so be careful.

If there are more files than can be shown on one screen display, you may advance to the next "page" by pressing **shift up-arrow**, or return to a previous "page" by pressing **shift down-arrow**.

The changes to the directory are maintained in a buffer in memory until you press **W** (for "Write to disk"). When you press "W", you will be asked to confirm your decision, i.e., whether you really want to write the changes back to the disk or not. This will give you a chance to change your mind. If you reply "Y", the updated directory is written back to the disk. If you reply "N", then all changes are cancelled, and the original directory on the disk is kept intact.

You may also input more than one drive number, separated by commas or spaces, when initially asked for the drive. Subsequently, pressing **A** will read in the directory of the next drive. If you have made alterations to the current directory, make sure that you have written it back to the disk before pressing "A" or all changes will be lost.

Throughout this procedure, BREAK, SHIFT-BREAK and F1 will remain active and will bring you to the PURGE Utilities menu or the main menu, respectively, if pressed. If F3 has also

been defined, it will also be active.

## II. KILL FILES BY CATEGORY

This option will permit you to remove certain classes of files from a disk with ease. You will be prompted for a drive, then for the common category. You may enter a filename extension, such as /BAS to remove all files with the /BAS extension, or /CMD to remove all files with the /CMD extension. Do not forget the initial slash ("/") if you are specifying an extension.

**WARNING:** On a TRSDOS 6.x disk, and on all non-RS system disks, if you specify /SYS, you will kill not only all system files, but also the BOOT/SYS and DIR/SYS entries in the directory as well! This is a very easy way to make a disk TOTALLY unusable! If you wish to remove the system files, use the wild-card characters option, described in the next paragraph, or the "Remove System Files" option.

You may also use a wild-card option to define a common category of files. Entering a letter or a group of letters alone will cause all files which start with that letter to be removed from the disk. For example, if you were to enter B, you would remove files with names such as BASIC/CMD, BACKUP/CMD, BULL/DOG, etc. If you entered BA, however, you would only remove the files BASIC/CMD and BACKUP /CMD.

Finally, you may remove classes of files based on their attributes. You may remove all INVISIBLE files, all VISIBLE files, all SYSTEM files, or files with a protection level other than 0 (no protection). However, see the caution regarding removing system files, above.

To remove a class of files based on their attributes, you must enter a SPACE as the first character when prompted for the common category, followed by I for invisible files, V for visible files, S for system files, or P for files with a

protection level other than 0. You may enter more than one attribute, separated by commas or spaces.

TRSDOS 1.3 and TRSDOS 2.7DD system files cannot be killed with this option, since these systems do not log them in the directory in the same fashion as normal files. See the next option.

## III. REMOVE SYSTEM FILES

This option is used to safely remove the system files on a diskette. The only input required here is the drive number (you may specify more than one, separated by spaces or commas). All system files except for BOOT/SYS and DIR/SYS will be removed.

Since the Super Utility 4 purge routines do not zero out the directory entries of removed files, these files may be reinstated as active files using the Restore command of KILL SELECTED FILES or the disk repair utility (see below). However, this is safe **ONLY** as long as no normal disk writes have been done to the disk (EXCEPTION: TRSDOS 1.3 and TRSDOS 2.7DD system files cannot be restored).

## IV. REMOVE ALL PASSWORDS

This routine will enable you to strip the passwords from all files on a disk, including invisible files and system files which are logged in the directory. You do not have to specify any previously-set password. When this routine is selected, you will be prompted for the drive numbers containing the diskettes from which passwords are to be stripped. The routine will then read into memory the directory track of each disk in turn, strip the passwords from each file, and change the protection level of each file to 0 (full access). Other file attributes will not be touched. The directory track will then be written back out to the disk, and the routine will

proceed with the next one, if any.

This routine will set the DISK master password to "PASSWORD." File passwords (both access and update) will be removed.

## V. DISK DIRECTORY

This option will, upon input of the drive number(s), supply you with the name and date on the disk, the number of tracks it was formatted for, the number of free granules remaining, and the number of free file entries available in the directory. In addition, it will display all valid files on the specified diskette(s). The display will include protection level (if any), and the file attributes (whether it is a SYSTEM file, an INVISIBLE file, or a visible file).

This option can work even if you do not know what DOS formatted the disk you wish to look at, by using a "!" in front of the appropriate drive number to activate Super Utility 4's special DOS detect routines. Note that this process may take a minute or two, and it should not be used on disks which have relative sectoring formats (e.g., ND 80 V2 double density) or disks which were created by operating systems not recognized by Super Utility 4.

For example:

**SYS0/SYS SIP-7**

means that the file SYS0/SYS is a system file (S), has the invisible attribute (I), and has a protection level of 7 (no access). The protection levels for TRSDOS 6.x, and for all other operating systems, and their meanings are as follows:

| level | TRSDOS 6.x | All others |
|---|---|---|
| 0 | FULL access | FULL access |
| 1 | REMOVE + 2,3,4,5,6,7 | KILL + 2,4,5,6,7 |
| 2 | RENAME + 3,4,5,6,7 | RENAME + 4,5,6,7 |
| 3 | WRITE + 4,5,6,7 | Unused |
| 4 | UPDATE + 5,6,7 | WRITE + 5,6,7 |
| 5 | READ + 6,7 | READ + 6,7 |
| 6 | EXECUTE + 7 | EXECUTE + 7 |
| 7 | NO Access | NO Access |

You will note that each protection level implies all the ones below it. In non-TRSDOS 6 systems, protection level 3 is unused and protection level 4 is what level 3 is on TRSDOS 6. You should not assign a protection level of 3 to any files on a non-TRSDOS 6.x disk because the results would be unpredictable. Protection level 7 is assigned to system files and means that the user has no access to those files.

If the disk directory has more entries than can be displayed on a single screen page, Super Utility 4 will pause at the end of the first screen page. Pressing any key will display the next page. If more than one drive was specified at the initial prompt, the next drive's directory will be displayed upon completion of the previous one.

## VI. ZERO UNUSED ENTRIES

This option will zero out the directory entries for REMOVEd files. Under most operating systems, a killed file is not removed from the disk directory; its entry is left in place, but flagged as "inactive." This procedure will completely remove the entry of killed files from the directory. It requires only the input of the disk drives containing disks whose directories are to be worked on.

The usefulness of this procedure with TRSDOS 1.3 and TRSDOS 2.7DD is questionable since these systems automatically zero out directory entries when a file is removed.

## VII. ZERO UNUSED GRANULES

This option may be used to remove ALL traces of removed files from a disk, not just their directory entries. This may be desirable when security must be maintained at a high level. Input the drive number(s), and the routine will zero out all unassigned sectors on the target disks. Once this utility has been run, no data recovery of any kind is possible for removed or inactive files.

This facility will not touch the diskette directory. To remove inactive file entries, you must use the ZERO UNUSED ENTRIES option. In most cases (with the exception of TRSDOS 1.3 and TRSDOS 2.7DD) when a file is removed the data remains on the disk, and so does the file directory entry; all that happens is that the file directory entry is flagged as inactive. To remove the file directory entry completely, it will be necessary to use the ZERO UNUSED ENTRIES option. To remove every trace of a file from a disk, use ZERO UNUSED GRANULES, along with ZERO UNUSED ENTRIES.

## VIII. CHANGE DISK NAME

Some operating systems do not supply an easy facility for changing the names or attributes of diskettes. This option in Super Utility 4 will permit you to do all that. You will be prompted for the drive number(s). The directory will be read, and the DISK NAME will be displayed and you will be prompted for another one. If you merely press ENTER at this point, the old name will be retained.

Next, the creation DATE of the disk will be displayed. You may supply a new date, or press ENTER to retain the existing

one. You will then be prompted for a new disk master password the existing password is NOT displayed). If you hit ENTER at this point, the password will be changed to "PASSWORD."

## IX. CHANGE FILE PARAMETERS

When you select this option, you will be prompted for a filename. After you supply one, the directory of the specified disk is searched for that file. If it is found, you will then be prompted for the new filename. Press ENTER to leave it unchanged. Next, you will be asked for the new USER ("access") password. Pressing ENTER here will set it to blanks. You will then be asked for the new OWNER ("update") password, and you can either supply one or also just press ENTER to set it to blanks.

The routine will prompt you for the protection level next. You must enter a number from 0 to 7 (see the table above).

Finally, you will be asked to supply the file attributes. Enter V if you want the file visible, I if you want it invisible, or S to declare it an invisible SYSTEM file. Note that any file can be given the SYSTEM attribute. At this time the updated information will be written back out onto the diskette.

When copying files to or from a TRSDOS 1.3 diskette to any other diskette, this routine will come in handy for removing the passwords from a specific file. TRSDOS 1.3 uses a different password encoding algorithm from all other systems, so a file which may not have passwords on one system is sure to have them when moved to the other! This routine will permit the removal of passwords from specific files only, if you do not wish to use the Remove All Passwords utility.

## X. CHECK DIRECTORY

This option scans the directories on the specified drives for errors. You will be advised of the disk name, date, number of formatted tracks, number of free granules and number of free directory slots. In addition, you will be advised of any errors in the directory. Such errors may include granules allocated to nonexistent files, HIT table entries which do not have corresponding file entries, or vice versa, or improperly linked extended directory entries. If any such errors occur, you may repair the disk by selecting the disk repair utilities of Super Utility 4 (see below).

```
** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **
  Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved

__ Repair Utilities __

 1 _ Repair GAT Sector          6 _ Recover Killed Files
 2 _ Repair HIT Sector          7 _ Move Directory
 3 _ Repair BOOT Sector         8 _ Display Directory
 4 _ Read-Protect Directory     9 _ Check Directory
 5 _ Un-Read Protect Dir

Choice? #__
```

## CHAPTER 4 - DISK FORMAT Utilities

Bring up the main menu by pressing SHIFT- BREAK. Now press 3 and ENTER. You will be presented with the DISK FORMAT Utilities menu:

```
** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **
  Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved

__ Format Utilities __

 1 _ Standard Format          4 _ Build Format Track
 2 _ Special Format           5 _ Write Format Track
 3 _ Format Without Erase     6 _ Software Bulk Erase

Choice? #__
```

This group of routines permits you to format your diskettes in a variety of ways. All of the utilities will produce formatted tracks that are readable by your particular operating system, except for option number 5, SOFTWARE BULK ERASE, which completely removes all data from a disk.

## I. STANDARD FORMAT

This option will format your target disk using the standard format for your operating system. You will be prompted for the drive number(s), the NAME for the disk, the DATE, and the MASTER PASSWORD to be used on the disk. If you press ENTER to the prompt for the name and date, Super Utility 4 will automatically supply the name "* Data ** Disk *" overlapping in the two (name and date) fields. Pressing ENTER in response to the PASSWORD prompt will cause "PASSWORD" to be used.

Finally you will be asked if you wish to use the CONFIGURATION for which the system has been set. A full discussion of the configuration tables is in Chapter 1. Essentially, you may configure Super Utility 4 to recognize that certain types of diskettes will be found in each drive. For example, you may set the configuration table to recognize that drive 0 will always contain a Model 4 TRSDOS disk formatted for 40 tracks, while drive 1 will always contain a single-density MODEL I DOSPLUS disk formatted for 35 tracks, and so on.

If you answer "Y" to the "Use Configuration ?" prompt, Super Utility 4 will scan the configuration data for that drive to determine how the disk should be formatted. The DOS type, number of tracks, and directory location values as currently entered in the configuration table will be used. If you answer "N", you will get additional prompts asking you to define the format further.

There is a very easy way to add tracks to a diskette using the standard format. For example, if you wanted to turn a 35-track disk into a 40-track disk, without losing any data that is already on it, just place the disk in a disk drive and select standard format. Go through all the prompts until you arrive at the **Use Configuration ?** prompt. Now reply "N" and press ENTER.

A further prompt will appear, as follows:

**:d Type,Tks, Dir, St Tk ?**

where :d is the target drive. Answer the prompts. Make sure you give it the proper DOS type. Give "40" for the track count, "17" for the directory track and "35" for the St Tk (Start track).

Super Utility 4 will then check the disk for the presence of valid data, and ask you whether you wish to continue or quit. If you elect to continue with the format at this point,Super Utility 4 will proceed to format the disk, starting at track 35 and moving up to track 39. You will then be prompted whether you want to write the directory track and boot sector. This prompt will appear only if the specified starting track number was other than 0. Reply "N."

**DO NOT REPLY "Y"** or a blank directory will be written to your disk, rendering previous files on the disk inaccessible!

To complete the process, check the configuration tables and make sure that the configuration for this drive correctly reflects the number of formatted tracks on the disk. Then go into the REPAIR Utilities menu and select the REPAIR GAT TABLE option and tell Super Utility 4 to repair the entire sector. This will open up the extra tracks on your disk, and your 35-track disk will be turned into a functional 40 track disk with 5 extra tracks of space.

This presupposes, of course, that your disk drive was capable of reading and writing 40 tracks in the first place. Also, in order to make use of the added space, your DOS must be capable of recognizing that it is there and be able to make use of it. Adding new tracks to a 35 track disk with TRSDOS 2.3 will not help you one bit, since TRSDOS 2.3 cannot recognize more than 35 tracks on a disk.

During the format, Super Utility 4 will first scan the disk in the target drive. If it sees something other than a blank disk, it will immediately display the message,

**DRIVE n has DATA!**

where n is the drive you specified for the Format operation. If possible, it will also display the disk's name and date. You will then be prompted whether you want to continue or abort the operation. If you elect to continue, the data on the disk will be overwritten. A request to abort will put you back in the DISK FORMAT Utilities menu.

There is one special override command which you can use when the prompt, "Use Configuration?" is displayed. Typing ! will force an immediate format; that is, Super Utility 4 will check Track 0 for data, but will not check the directory if it does find data, and will not prompt you whether it should continue or quit, proceeding with the format instead. Note than when you are formatting several disks at the same time, this lack of prompting will carry over to the other drives as well!

When formatting is completed, you will be asked, "Repeat?" You may now take the disk from the target drive and insert a fresh one. Then, typing "Y" will cause the format operation to be repeated. The same information as to DOS type, number of tracks, etc. will be used. If you wish to format a disk differently, you must hit BREAK and re-select standard format in order to specify the new configuration. Typing "N," however, will return you to the DISK FORMAT menu.

If during the format process bad granules are encountered, these will be mapped into the track lockout table in the GAT sector of the directory. However, if you are formatting a disk which uses a relative sector format scheme (i.e, NEWDOS80 V2 double density or MULTIDOS P-Density) there will not be room for mapping bad granules in the GAT table. In fact, these types of disks do not use a lockout table in the same way as

regular disks do. What Super Utility 4 will do in such a case is allocate the bad granules to prevent their use by the operating system. The user should note that this procedure of allocating bad sectors may result in subsequent Directory check errors. They should NOT be fixed!!

## II. SPECIAL FORMAT

This routine allows you to construct one or more tracks sector by sector, with a variety of formatting options. You can control the contents of the identification fields for each sector, the data address mark for each sector, and even the placement of false CRC bytes. In addition, you can construct a track with a mixture of single and double density sectors This routine may be used for constructing specially-formatted "protected diskettes."

When you enter this routine, you will first be asked how many tracks you want to create. After you have entered a number, you will be shown a line that looks like this:

**TRACK 00 :SINGLE 00 DOUBLE 00**

with a prompt line beneath it. This line indicates which physical track and sector you are working on, and whether the sector is single or double density. The position of the colon (:) indicates which it is. If you are going to mix densities within a track, you must construct all your single-density sectors first before going on to the double-density sectors. Once you start on a double-density sector there is no going back to single till the next track.

To construct an individual sector, you must supply the routine with the necessary information for that sector. These are:

| | |
|---|---|
| Tnn | TRACK number |
| Hnn | HEAD number |
| Snn | SECTOR number |
| Lnn | SECTOR length |
| Ax | DATA ADDRESS MARK type |
| Cy | CRC type (true or false) |

T, H, S, and L may be followed by a number, or the letter R. That is, you may specify which track, head, and sector number, and sector length, is to be associated with this physical sector. If you specify R instead, you will generate random numbers for the particular parameter you give R to. Any parameter not specified will default to certain values. Track number will default to the physical track number. Sector will default to 0. Head number defaults to 0, length defaults to 1. Remember that the track and sector numbers do not have to correspond with the physical track and sector numbers! In fact, each sector on a track may have a different track and sector number associated with it! HEAD refers to a byte which indicates which side of the disk this sector is on (Head 0 indicates the first side, Head 1 indicates the second side. You can use anything.).

The L parameter is a coded value which indicates the number of data bytes in this sector. It conforms to conventions, i.e., 0 would mean a 128-byte sector, 1 indicates 256 bytes, 2 indicates 512 bytes, and 3 indicates 1024 bytes.

The type of data address mark to be used for this sector only may be specified with the letter A followed by one of four identifiers: S for standard DAMs, or R for read-protected. For example, a read-protected data address marks would be specified by AR.

The condition of the CRC bytes can also be controlled by entering the letter C followed by a number from 0 to 3. The numbers correspond to certain conditions, given below:

| | |
|---|---|
| 0 | Normal ID and DATA CRCs (default) |
| 1 | Correct ID field CRC, error on data CRC |
| 2 | Correct ID field CRC, no data CRC or sector header written |
| 3 | CRC error on ID field, no sector header or data CRC written |

Thus, to create one sector, a typical command line might be:

**T21 H0 SE0H L1 AR C0**

This line creates a single sector with the following information: The sector has a track number of 21, has a head ID of 0, a sector number of E0H, a length byte of 1, read-protected data address marks, and normal CRCs. The next sector might be created using **TFFH H1 S8FH L0 AS C1** which creates a sector with a track number of FFH, and has a head id of 1, a sector number of 8FH, standard data address marks, a correct ID field CRC but false data CRCs.

To switch from single to double density, enter a D on the prompt line as the first character of your sector specifications. This indicates to the routine that this sector and all following sectors created are to be in double-density (D needs to be entered only once). Remember that you cannot return to producing single-density sectors once you have moved to double-density until you move to the next track.

Pressing ENTER on a blank prompt line will terminate all processing for the current track. Your special format track will now be residing in memory. You will now be presented with a new prompt asking if you wish to examine the track before writing it out to disk. If you reply "N", then the tracks will be written out to the destination disk.

If you elect to examine the track, you will be placed in the DISPLAY MEMORY mode and the track buffer will be displayed. You can now load up its sectors with data, and make any alterations you wish. Pressing the spacebar will write one track to the disk and return to displaying the NEXT track in the buffer. Thus each press of the spacebar will write one track out. Pressing CLEAR will write all the sectors out to the destination disk without returning to the display mode. The track will be written to the disk, and the program will advance you to the next physical track. As usual, pressing BREAK, shift-BREAK or F1 will terminate the procedure and return you to the FORMAT and main menus, respectively.

While in the track display mode, all DISPLAY MEMORY paging and modification controls are active as usual (except for the CLEAR key). See the MEMORY UTILITIES chapter for more information.

## III. FORMAT WITHOUT ERASE

This routine will permit you to reformat a disk without losing any readable data that was already on it, and is very useful for revitalizing disks which have been lying around for a long period of time. It is also a good way to repair "Sector not found" and "Data CRC" errors. You will be prompted for the drive number(s). If you specify a "!" at the initial prompt for the drive numbers Super Utility 4 will scan the disk to see how it is formatted, then proceed to reformat it without destroying any data.

When you press ENTER to start the routine, Super Utility 4 will scan each track, reading each sector into a buffer in memory. It will then reformat that track, and write the data back out to it.

If during this process Super Utility 4 encounters a bad sector, it will stop and present you with the R>etry, S>kip, C>ontinuous, N>onstop or Q>uit ? menu. If you are trying to

recover a bad disk through this method, use R, C or N as heavily as possible. Don't skip a bad sector at once. If Super Utility 4 can read the sector at all, it will rewrite it back out with the correct format, thereby repairing the error. So give the program a chance. It may take several minutes, but if the sector can be read at all, Super Utility 4 will do it -- and recover your data.

## IV. BUILD FORMAT TRACK/WRITE FORMAT TRACK

These two routines will permit you to create specific tracks and write them out to disk at specific locations. This may be useful for saving a glitched diskette, by replacing the damaged tracks with new ones. The data on the damaged track(s) will be lost, of course, but the rest of the disk will be intact.

The BUILD FORMAT TRACK procedure differs from Special Format in that it only constructs standard format tracks and requires no input from the user other than the DOS type and track number. When you select BUILD FORMAT TRACK, you will first be asked to specify the DOS type. Reply with one of the valid DOS type identifiers to create the appropriate track. You will then be asked what track number to assign to it. This should be the track number where you intend to write it out to the disk.

Super Utility 4 will then create the track in memory, and display the message, "Press <ENTER> to view the track at nnnnH". By pressing ENTER, you will be able to view the track buffer in memory. Press BREAK to return to the DISK FORMAT Utilities menu.

Once you have built a format track in memory, you may write it out to disk by selecting WRITE FORMAT TRACK and specifying the track number. The track in the memory buffer will be placed on the disk. Note that you can build a track in memory with one track number, but write it out to another

location on the disk. This is entirely possible, but not recommended.

## V. SOFTWARE BULK ERASE

This utility will completely remove all data from a target disk and can be used to clean up a disk in place of a bulk eraser or magnet. If you have problems with the DOS rejecting a backup or a format due to "different pack IDs," this routine will solve that problem. You will be asked which drives are to be bulk-erased. Make sure they contain the proper disks! When you press ENTER, the disk will be overwritten completely with 00 bytes, removing all traces of formatting and data. Take as much care in using this routine as you would in using a bulk eraser, since there is no recovery possible.

# CHAPTER 5 - BACKUP UTILITIES

There are two routines to this group: Standard Backup and Special Backup. Standard Backup is used on those disks which can be identified using one of the valid DOS specifiers. Special Backup is used for all others.

## I. STANDARD BACKUP

Super Utility 4 has a very fast backup routine which can be used in place of the standard BACKUP/CMD of your operating system(s). It will work on a one-drive system as well as on a multi-drive system. Additionally, more than one destination drive can be specified, in which case, the source disk will be backed up to the other drives one at a time.

When selecting this procedure, you will first be asked for the source drive. Enter the drive number which contains your source disk. Do not enter more than one source drive. You will then be asked for the destination drives. You may enter more than one, separated by commas, or spaces.

You will next be asked if you wish to format the destination drives first. You may reply "Y" to format, "N" if the disk is already formatted, "!" to force a format even if the disk has data, or "*" to format and skip the verification cycle. If you type "Y", "!", or "*", each destination drive will be formatted. To avoid any problems, make sure you have configured both the source drive and the destination drive(s) with the same DOS specifier. While it is possible to back up one disk onto another with a different DOS specifier, this is not recommended (see below).

Immediately after formatting, a full disk backup will be performed. Unlike the backup procedure of some systems, this backup will not skip any empty tracks. Nonetheless it will

still be quite fast.

If you do not wish to reformat the destination disks, merely reply "N" and the backup will proceed. The destination disks must of course have previously been formatted.

After the source disk has been backed up to all the destination drives, you will be asked "Repeat?" Reply "Y" if you wish to do more backups, or "N" to return to the menu.

If you backup one disk to another with a higher track count, as for example a 35-track disk to a 40-track disk, the backup disk will still reflect the availability of 40 tracks rather than 35 as would result from using other Backup programs.

When using the backup procedure to copy disks, the source and destination diskettes must have a compatible configuration. For example, backing up a single-sided disk onto a drive configured as double sided is illegal, and will result in an error.

Also, the drives must be of compatible type. Attempting to back up a double-sided disk placed in a single-sided drive will result in severe errors. Similarly, backing up a TRSDOS 1.3 disk onto a drive configured for TRSDOS 6.x double-density or any other disk to a drive configured for a different DOS will result in a badly messed-up disk. Remember that by definition BACKUP implies the creation of an EXACT copy of the source disk.

The BACKUP utility is the easiest way to recover the data from a disk with bad sectors. As each bad sector is encountered, Super Utility 4 will stop and present you with the **R>etry, S>kip, C>ontinuous, N>onstop or Q>uit** prompt. Use R, N or C whenever this happens. If the sector can be read at all, it will be copied over intact to the backup disk, thereby recovering the data. Even if not all of the sectors can be copied, you will still wind up with a usable (that is, totally

readable) disk from which you can copy off as much useful data as you can. In addition, the original disk, bad sectors and all, will still be available for another try.

**WARNING:** Do not use the SKIP option when you have multiple destination drives. Any disks further down the line will not be copied to properly. If you run into a bad disk during a multiple-destination backup, it is best to abort the procedure and start out again with a fresh set of disks.

## II. SPECIAL BACKUP

The special backup routine is intended for disks that do not conform to any of the valid DOS specifiers. This are usually disks with damaged formatting information, or a disk deliberately formatted in a non-standard manner. This routine requires the input of the number of physical tracks on the target disk (or a reasonable estimate thereof). When ENTER is pressed, the program will begin scanning the source disk and building a table in memory of its formatting characteristics. This will take several minutes. If the routine is able to successfully scan the entire disk, it will then ask you whether you want to view the format track buffer in memory. At this point, Super Utility 4 has constructed an image of the source disk's formatted tracks in memory, and you can reply "Y" to display them (in DISPLAY MEMORY mode) to examine their structure. Note that these are bare formatted tracks, with no data in the sectors. Pressing the spacebar will write the current track to the destination disk and return to displaying the NEXT track. Pressing CLEAR will write the buffer to the destination disk without returning to the viewing mode. While in the viewing mode, all DISPLAY MEMORY paging and modification controls (except for CLEAR, which has a different function) are active. See the MEMORY UTILITIES chapter for more information.

If you do not wish to view the formatted tracks in memory, simply reply "N" in response to the prompt, and the tracks will be written out to the destination disk. After the destination disk has been formatted, then the data will be copied from the source disk onto the destination disk.

This routine is intended to be a last-ditch recovery procedure for damaged disks which the standard repair methods cannot handle. It can also handle many disks formatted in a non-standard manner, however there is no guarantee made that the Special Backup routine will be able to duplicate every disk ever produced with a special format. In fact, there exist particular formats which by their very nature cannot be easily duplicated, or duplicated only partially, without introducing unwanted changes into their structure. There are also certain types of formats which, while they can be duplicated, cannot have any data written to them without destroying their special characteristics. If you encounter a special-format disk which cannot be copied with this routine, please contact your dealer or supplier to obtain a copy. Most dealers who sell protected disks have a reasonable backup policy and will allow you to purchase a copy at a nominal fee. Please do not call us for updates to the SPECIAL BACKUP routine.

# CHAPTER 6 - REPAIR UTILITIES

Selection of this group from the main menu will produce the following menu display:

```
** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **
   Copyright # 1983, 1984 Breeze/QSD, Inc. - Dallas, TX

__ Repair Utilities __

 1 _ Repair GAT Sector          6 _ Recover Killed Files
 2 _ Repair HIT Sector          7 _ Move Directory
 3 _ Repair BOOT Sector         8 _ Display Directory
 4 _ Read-Protect Directory     9 _ Check Directory
 5 _ Un-Read Protect Dir

Choice? #__
```

These utilities are designed to restore disks which might be unusable or contain garbled data to a usable condition, if it is at all possible. Many times a disk becomes unreadable because of some damage to the directory. If the damage is not too extensive, the REPAIR utilities will be able to fix it. However, the disk itself must be readable, i.e., it must not contain any format errors or CRC errors which will prevent Super Utility 4 from reading it. If necessary, use FORMAT WITHOUT ERASE or the backup procedure described in the previous chapter to make sure the disk is readable. Note also that the REPAIR utilities assume that there is no data loss or

damage in the third and following sectors of the directory, where the file entries are maintained. If there is data loss in this area it may not be possible to fully recover a damaged directory.

The REPAIR utilities also contain routines which will allow you to check on the condition of a directory, recover any files killed by the PURGE utility of Super Utility 4, move the directory track from one location to another on the disk, or clear the unused directory entries by setting them to zero.

## I. REPAIR GAT SECTOR

The **GAT** (Granule Allocation Table) sector of a diskette directory is where the operating system keeps track of which granules have been assigned to files, which are available for use, and which have been locked out for one reason or another. An error in the GAT can result in files being allocated disk space which already belong to other files, thus resulting in data loss. This is the kind of creeping error which can eventually render an entire disk worthless.

The GAT REPAIR option will permit you to repair a bad GAT table. Upon entering this option, you will be asked for the drive number containing the disk to be repaired. You may enter more than one drive number if you are repairing several disks. You will then be asked whether you want the GAT Table only (the first 96 bytes), or the entire sector, to be repaired.

If the disk name and date are unreadable, select the ALL option to rebuild it. Otherwise, if the GAT table alone is damaged, select GAT Table repair only. This will allow you to retain the original disk name, date, password, and any active auto command. Super Utility 4 will then proceed to rebuild the granule allocation table of the target disk(s) by reading the allocation information for each file in the rest of the directory.

TRSDOS 1.3 for the Model III as distributed by Tandy has a built-in error in its GAT table. This appears to be the result of two files which were improperly killed. The allocation bits for the third and fourth grans of track 2 appear in the table as allocated, although no directory entries for them exist. REPAIR GAT may be used on such disks and will free up the two mis-allocated granules for user files. This error has been present since TRSDOS 1.3 was first issued, and has never been corrected. Granule 3 of Track 2 appears to have been a program to read standard format disks (sectors 0-17) into TRSDOS 1.3 format (sectors 1-18), and Granule 4 appears to have been a one-byte data file. What these two files were used for is unknown.

## II. REPAIR HIT SECTOR

Each filespec in a directory is "hashed" by the operating system into a one-byte code and entered into the HASH INDEX TABLE sector, in a position which defines the file's actual position in following directory sectors. This permits the operating system to find filespecs in the directory very quickly. Errors which may occur here are HIT codes assigned to nonexistent files, invalid HIT codes, etc. Or the sector itself may have been damaged so that part or all of it may have been turned into garbage.

If a directory check reveals errors in the HIT sector, you may use this option to repair it. The only required input is the drive number containing the damaged disk (you may enter more than one drive number). When you press ENTER, Super Utility 4 will proceed to reconstruct the entire HIT table.

TRSDOS 1.3 codes the system files in the last 32 bytes of the HIT table, and TRSDOS 2.7DD in the last 48 bytes. Because of the fact that they do not have any standard directory entries, these bytes will be left untouched by Super Utility 4 during a HIT repair, since without standard entries, it is not possible to reconstruct a HIT entry. If these bytes have been

corrupted, you must use the ZAP utility to effect repairs. If the bytes look obviously invalid to you, enter FF in all 32 slots (48 for TRSDOS 2.7DD).

## III. REPAIR BOOT SECTOR

If a system disk will not boot, the chances are that the boot sector has been damaged. This sector contains code which loads the rest of the operating system into memory, so if this sector is corrupt, the rest of the DOS will not load. If this is the case, you can effect a repair of the boot sector by selecting this option. You will be prompted for the drive number containing the disk with the damaged boot. When you hit ENTER, Super Utility 4 will write the new boot routine into this sector.

Super Utility 4 contains three different boot routines, which are intended for use on **TRSDOS 2.3, 1.3, or 2.7DD disks** ONLY, and which Super Utility 4 can use to replace a damaged boot sector. These boots are for a single density disk, a double-density disk with a single density Track 0, and for a pure double-density TRSDOS 1.3 disk. The one which Super Utility 4 writes to your disk will depend on the DOS type assigned to that drive in the Configuration table. If you attempt to use this procedure on any system other than the three listed above you will get the message,

**USE COPY SECTORS FOR THIS DOS -- SEE MANUAL**

This means that instead of trying to use the default boots, you should copy the boot sectors from a good disk onto the damaged one. If the problem lies in the boot sectors, then first make sure the boot track is readable (no CRC errors, etc.). You may need to reformat that track to assure this. Then copy the first two sectors of Track 0 from a known good disk onto the corresponding location on the bad one. This is done by going to the ZAP utilities and selecting option 4, COPY SECTORS. Consult Chapter 2 for more details.

The default boots in Super Utility 4 are not the same as the standard boots, but very fast routines which check each byte of the the resident operating system module (SYS0) as it loads it into memory to ensure that the load is a good one.

This repair option does not log the boot in the directory (the operating system will already have it logged in anyway). All it does is write the new routine into the sector formerly occupied by the damaged one.

## IV. READ-PROTECT DIRECTORY

If you try to boot a system diskette, and the disk drive seems to have difficulty in finding a file, or if you try to read a file on a data diskette and the drive seems to be hunting all over the diskette, or if you try to display a directory and the disk drives grind along for several minutes, the chances are that all or part of the directory was re-written using the wrong DATA ADDRESS MARK (Note: this can also happen if you try to read a single-density Model I TRSDOS diskette on a Model 4 or a Model III machine). The directory track of a diskette has a data address mark (see chapter 2) which is different from all the other tracks, and is used by the operating system to identify its location.

Whenever the DOS tries to read a directory and finds that some or all of the directory sectors do not have the correct data address mark, it will hunt all over the disk for a track that has the proper DAMs. It will not find it, and so your drive sounds like its head keeps moving back and forth (which is exactly what is happening).

The term "read-protecting a directory" is a misnomer and refers to the process of writing a directory track with the correct data address mark which can be recognized by the operating system. It does NOT mean changing the directory so that it cannot be read at all. Unfortunately, the term stuck.

However it should NOT be taken literally.

When this option is selected, you will be prompted for the drive number(s) containing the target disks. When you press ENTER Super Utility 4 will read the directory track, then write it back out with the correct data address marks. If it cannot find the directory you will be asked to specify its track location and length (i.e., number of sectors in the track). You may enter the track number where the directory is located followed by its length, or press ENTER to default to the directory track number and track length in the configuration table.

If you recall the discussion of data address marks in chapter 2, you will realize that if you try to read a MODEL I single-density TRSDOS disk in a Model 4 computer, your Model 4 (or Model III) will not be able to find the directory track. The reason is that the data address mark used by Model I TRSDOS (and other similar systems) is one that the Model 4/4P (and Model III) hardware cannot recognize. If this problem does not occur with another operating system, it is most likely because that system uses a data address mark for the directory track that both the Model I and Model 4/4P can recognize.

This has other implications. Super Utility 4 will adapt itself to whatever machine it is running on. If you attempt to read-protect a TRSDOS 2.3 disk directory on a Model III machine, you will write the WRONG data address marks on the Model I disk! So even after you thought you had fixed the problem, now your TRSDOS 2.3 will be unable to locate the directory track!

If you are going to use this option, therefore, make sure that you are using it on the RIGHT machine. Do not attempt to read-protect a TRSDOS 2.3 directory track on a Model 4 computer, and vice versa. This restriction may not apply to other operating systems, however.

## V. UN-READ PROTECT DIRECTORY

This option takes a directory track and rewrites it using standard data address marks (or RPT data address marks, if the disk is Model III TRSDOS 1.3). The only input required is the drive number. You may want to use this routine if you are developing a protected disk with a directory that you do not want any standard operating system to be able to read.

## VI. RECOVER KILLED FILES

This routine will allow you to recover files previously killed using Super Utility 4's PURGE utilities. You cannot use it to recover files killed with the "KILL" command used by TRSDOS 1.3 (Model III) or TRSDOS 2.7DD (Model I) , as the entire directory entry is zeroed out when a file is "KILL"ed (you may, however, use it to recover a file REMOVEd under TRSDOS 6.x and other non-RS operating systems).

Remember that recovering a killed file is a chancy affair IF you have performed any disk writes since the time you killed the file. In the first place, you may introduce errors into the directory's GAT (granule allocation table). Secondly, the file you recover may have had parts of it overwritten by subsequent files. So be careful when using this routine.

You will first be prompted for the drive number or numbers. Upon hitting ENTER, Super Utility 4 will display the disk name, date, number of formatted tracks, and number of free granules and free directory slots on the disk. Press ENTER again. You will now be shown a list of all VALID and NON-VALID (i.e., killed) files in the directory of the target disk. All valid files will be surrounded by right and left arrows (right and left square brackets on the Model III). Non-valid files will be surrounded by solid graphics blocks.

To recover a killed file, move the cursor to the file that you wish to restore, using the arrow keys. Press R and you will see the graphics blocks replaced by right and left arrows (or brackets), indicating that the file is now a valid directory entry.

When you have recovered all the files you want to, press W to write the updated directory back onto the disk. You will be asked to confirm your decision. Press "Y" to confirm and write the directory, or press "N" to abort the procedure. Pressing A will abort the process and proceed to the next drive if more than one was specified, so if you have made any changes, make sure you write the directory back out to disk before using A to proceed to the next target drive.

If there are more files than can be displayed on the screen at one time, you can press SHIFT and UP ARROW to go the next display page, or SHIFT and DOWN ARROW to move to the previous page.

## VII. MOVE DIRECTORY

This option will move the directory track to any other track on the target disk as long as that track is not already allocated to another file. You will be prompted for the drive number(s) containing the target disks, and the track number where you want the directory moved. If the track you selected is already allocated (either fully or partially), Super Utility 4 will display a message to that effect and ask you to select another track. If you press ENTER without giving it a new location, Super Utility 4 will display the available tracks where the directory can be safely moved. Upon entering the track number and pressing ENTER the directory track will be moved, and the boot sector changed so that it now points to the new updated directory (the position of the directory is coded in the second or third byte of the boot sector depending on the operating system involved). The old location will be rewritten with the correct DAMs.

On TRSDOS 6.x (Model 4), TRSDOS 2.3 (Model I) and all other non-RS disks, the DIR/SYS entry in the directory will be updated to reflect the directory's new location.

If you wish to abort this routine, BREAK, SHIFT-BREAK and F1 will always take you back to the menus. So will the user-defined F3 key.

## VIII. DISPLAY DIRECTORY

This option requires only the drive number or numbers containing the disks whose directories you wish to view. It will give you a full screen display of all active files on the disk(s), along with the attributes and protection levels of each file (protection level 0 is not displayed). In addition, the disk name, date, number of free granules and free space in K (kilobytes, units of 1024 bytes), and number of free directory slots are also displayed.

If you view a TRSDOS 1.3 or 2.7DD directory, you will not see any system file names displayed. As was discussed above, TRSDOS 1.3 and 2.7DD does not code the system files in the directory in a normal manner. They will therefore be displayed as numbers, e.g.

**00-01-02-03-04-05-06-07-08-09-10-11-12-13-14**

indicating which of the relative system files are currently active (not killed) in the directory.

## IX. CHECK DIRECTORY

This option is designed to perform a very thorough check of the directory on the target disk. You will be prompted for the drive number. When you hit ENTER, the routine will scan the directory. It will then display the disk name, date, the

number of free granules and the number of free directory slots. Any errors encountered will then be displayed.

If the routine reports any GAT or HIT errors, you may use the GAT REPAIR or HIT REPAIR options to automatically repair the directory.

# CHAPTER 7 - MEMORY UTILITIES

The MEMORY Utilities of Super Utility 4 permit you to view and manipulate memory in a variety of ways. Many of these routines are very similar to the ZAP utilities, as can be seen from the menu below:

```
** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **
  Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved

__ Memory Utilities __

 1 _ Display Memory          7 _ Test Memory
 2 _ Move Memory             8 _ String Search
 3 _ Exchange Memory         9 _ Input Byte from Port
 4 _ Compare Memory         10 _ Output Byte to Port
 5 _ Fill Memory            11 _ Track to Memory
 6 _ Reverse Memory

Choice? #__
```

In addition, the memory utilities permit you to read and write to any I/O port from 0 to 255. You may also perform a direct memory-to-disk or disk-to-memory transfer.

Great care should be taken when modifying memory in any of the ways made available to you from this menu. Unlike the modification routines in the ZAP utilities, changes to memory are not buffered and take effect immediately on entry.

## I. DISPLAY MEMORY

This option requires input of the starting memory address that you want displayed. You may enter the address in any of the four number bases that Super Utility 4 will accept (Hexadecimal, Decimal, Octal, or Binary). The default base is Decimal. To enter a hexadecimal number, you must append "H" to the number; to enter an octal value, append O (the letter "O" -- not zero!!!) or Q and to enter a binary number, append B to the number.

The display is very similar to the ZAP display, with the exception that the first column of information is empty, and the actual addresses (in hex) are displayed in the second column, adjacent to the actual data. The data is displayed in blocks of 256 bytes each. You can look on this display as a window into the TRS-80's memory which can be moved around so as to present different views of the RAM or ROM "landscape."

Using the arrow keys, you can scroll through memory. By pressing the right or left arrows, you can move the display window one byte at a time in either direction. Pressing SHIFT-right arrow or SHIFT-left arrow will advance or decrease the display 256 bytes at a time.

Pressing up-arrow or down arrow will also move the display window 256 bytes at a time in either direction. However, pressing SHIFT-up arrow has a different effect; this will cause the highest 256-byte block of memory available in the computer to be displayed. Similarly, pressing SHIFT-down arrow will display the lowest 256-byte block of memory.

Pressing the CLEAR key will cause Super Utility 4 to prompt for a new memory address to display.

Pressing M will put you in Modification Mode. As in the ZAP utility's modification mode, you can select your input base by pressing H (hexadecimal), O or Q (octal), D (decimal), B (binary) or A (ASCII) before pressing M. The cursor may be

moved with the arrow keys, and the action of the modification control keys are the same as in the ZAP utilities. See table 2-2 for the various keys and their actions.

In memory modification mode, however, if you attempt to move the cursor past the display window, more data will simply be brought in, one byte at a time. If you attempt to move the cursor past the highest address in the computer, FFFFH, the display will wrap around to display 0000H. Pressing ENTER in modify returns to paging mode.

## II. MOVE MEMORY

This routine enables you to move a block of memory from one location to another. You will be prompted for the starting and ending address of the block to be moved, and the starting address where it is to be moved to. The prompt will look like this:

**Start, End, Start ?** _____

If you wanted to move the bytes located at 7000H through 70EBH inclusive to a new location starting at AB44H, you would then reply as follows:

**Start, End, Start ?**7000H,70EBH,AB44H

and press ENTER. The routine will then execute the memory move, and you will be advised of the number of bytes moved.

Be careful when moving things around in the lower 37K of memory, which is occupied by Super Utility 4 itself! If you move a block of memory into a location being used by the program, you may destroy vital parts of it and find yourself facing a disastrous system crash.

## III. EXCHANGE MEMORY

This routine is similar to "Move memory," except that it actually exchanges the contents of the origin and destination blocks of memory instead of merely copying the contents of the origin block over what was in the destination block.

The prompts will be identical to Move Memory, and will request the starting and ending locations of the origin memory block, and the starting location of the destination memory block where the exchange is to take place. For example, if you wanted to exchange the contents of the block of memory in 8000H through 87FFH with the contents of memory at C000H, your reply would be

**Start, End, Start?**8000H,87FFH,C000H

and when you press ENTER, the two blocks of memory will be exchanged. The bytes which resided at 8000H through 87FFH now reside at C000H through C7FFH, and vice versa.

## IV. COMPARE MEMORY

This option permits you to compare one block of memory with another. Suppose you had executed a memory move using option 1 and wished to verify if the move was properly executed. You would then enter the starting and ending addresses of the original block and the starting address of the comparison block.

The program will do a byte-by-byte comparison of the two blocks of memory and advise you of any mismatches that it finds. If there are many such mismatches, they will scroll past on the screen at a high rate of speed, but the display may be paused by holding down the spacebar.

## V. FILL MEMORY

Memory may be filled with a single byte value using this routine. You will be p prompted for the starting and ending addresses of the block to be filled, and the byte value to use as the filler. For example, if you wanted to fill the memory locations from DA00H through DFFFH with the byte 0AH, you would enter, **DA00H,DFFFH,0AH.**

All of the address entries have default values. The starting address location defaults to the first byte beyond the end of the Super Utility 4 program itself, and the end byte defaults to the top of memory. The fill byte defaults to 00. Thus, pressing ENTER at this prompt will zero all memory not used by Super Utility 4 and can be used to "clean up" memory for other procedures which may require use of this space (such as "Track to Memory" -- see below).

Be careful when using this routine in the lower 37K of memory as you could wipe out a critical part of Super Utility 4 and cause the program to crash. You can easily determine the first free memory address by selecting Display Memory and hitting ENTER in response to the address prompt. Super Utility 4 will display the first 256 bytes of **FREE** memory. You should not perform any FILL MEMORY function below the address on the upper left hand corner of the display.

## VI. REVERSE MEMORY

This option performs the same function as the "Reverse Sectors" option of the ZAP Utilities, except that it works on a block of memory rather than on a disk sector, and the area to be reversed is not limited to 256 bytes. You only need to enter the starting and ending addresses of the memory block to be reversed, and the routine will execute immediately. On completion of the routine you will be advised of how many bytes had been reversed. Note that unlike Reverse Sector Data, which is sector oriented, this routine is block oriented and will

perform the reversal from the starting address to the ending address inclusive.

## VII. TEST MEMORY

This option will perform a test of the region of memory that you specify. The test is a very complete one, but will not disturb the previous contents of the tested addresses! After each byte is tested, its original contents are restored. Thus you could test the region of memory occupied by Super Utility 4 without destroying the program in memory.

There is, however, an exception to this. You cannot test the actual locations from which the routine executes, or it may malfunction and behave unpredictably. Thus, when you select this option, you will see the message,

```
DO NOT test between xxxxH and yyyyH !!
Start, End ?_____________________
```

xxxxH and yyyyH are the locations where the test routine executes from, and altering them at the wrong time will definitely cause unwanted results.

To use this routine, merely enter the starting and ending addresses of the block of memory you wish to test and press ENTER. The memory test may be stopped at any time by pressing the CLEAR key.

Also, if you test the video RAM on a Model I without a lower-case modification installed, those addresses (3C00H through 3FFFH) will also be reported back as being bad. However, this is normal (the video memory on an upper-case -only Model I only has 7 bits per byte instead of 8) and does not indicate that something is wrong with your video memory.

Any errors encountered will be displayed by this routine, and you will be given a bit-by-bit breakdown of the problem addresses.

## VIII. STRING SEARCH

This routine will allow you to search memory for the occurrence of a particular ASCII string, byte string, or two-byte word string, and replace them with another string of your choosing. The routine works in the same fashion as the STRING SEARCH routine of the ZAP Utilities, except that this one works on memory, not disk. You must enter the starting and ending locations of the memory block to be searched. Next you will be asked to enter the string to be searched for.

Enter an ASCII string directly. Surround the string with double quotes if you want an exact case match on the search, or with single quotes if you want a case independent search to be made. To enter a byte string, enter a series of values in the range 0 - 255 decimal and separate them with spaces or commas. The values may be entered in any of the acceptable numeric bases recognized by Super Utility 4. To search for two-byte WORD strings, enter two-byte values separated by spaces or commas. The two-byte values should not exceed the range 0 - 65535 decimal. Remember that the routine will reverse the order of two-byte words in order to correctly search out address references, which are kept in LSB-MSB (least significant byte - most significant byte) order. Also, note that leading zeroes may cause the program to evaluate a two-byte value into one byte. For example, 0033H would evaluate to 33H instead of 3300H.

As with the search string function of the ZAP utilities, you may enter a search string of intermixed ASCII, one-byte values, and two-byte values.

If you do not wish to replace the original string, merely press ENTER when prompted for the replacement string, and the

program will display the matches as it finds them, along with their locations. If there are many of them, they will scroll past at a high rate of speed. Hold down the spacebar to temporarily stop the display. If the string was to be replaced, they will be replaced at this time. Remember that if the replacement string is longer than the search string, it will be truncated to the length to the search string. However, if the replacement string is shorter than the search string, only the corresponding number of bytes in the search string will be replaced.

## IX. INPUT BYTE FROM PORT

This option will allow you to input and display a byte from a hardware port. You need only supply the port number from which you wish to input. For example, if you wished to read the modem status register on your machine (port E8H) you would reply E8H (or 232 in the default decimal base) to the prompt, "PORT ?" and Super Utility 4 would immediately return the value in that port.

## X. OUTPUT BYTE TO PORT

Conversely, this option permits you to send a particular byte out to a hardware port of your choice. You will be asked for the PORT number and the byte you wish output to it. Upon hitting ENTER, Super Utility 4 will immediately send that byte out to the specified port. There will be no acknowledgement.

## XI. TRACK TO MEMORY

This option will read a full track of data into memory, including all formatting information such as header fields, CRC bytes, gap bytes, data address marks, etc. You will be asked to input the drive number and track number where you want to start examining. Then you will be asked whether or not

you want the floppy disk controller to synchronize to the ID marks during the track read. "Synchronizing to ID marks" means that the controller will start its accumulation of data at the address marks rather than elsewhere on the track. If you want this option, reply "Y," otherwise reply "N."

An entire track of information will be transferred from the disk into a holding buffer in memory, and you will be informed where it is located. You may then view the data by pressing ENTER. Doing so will place you in DISPLAY MEMORY mode, with all the operations described above available.

TRACK TO MEMORY will read only one side of a double sided disk at a time. You can specify which side to read by appending a single quote to the drive number (e.g., 0') for the front side, or a double quote (e.g., 0") for the back side. The side indicator **MUST** be specified if you are working with a double sided disk; leaving it off will cause TRACK TO MEMORY to read the back side of the disk. Also, if you elect to read the front side, you should be aware that this will cause Super Utility 4 to change the configuration data for that drive to reflect a single-sided drive. See Chapter 1.

Note that the TRACK TO MEMORY operation is not reliable when double-density tracks are involved. The reason for this is that when reading a double-density track, the controller chip automatically synchronizes to ID marks regardless of whatyou told Super Utility 4 to do. In other words, the instructions to not synchronize to ID marks is completely ignored by the chip. The problem arises in that certain bit patterns can fool the FDC into thinking that it is looking at an ID mark rather than real data. The FDC will then resynchronize that that point. The net effect is **lost data**.

An example of this is a track with the word "TRSDOS" in upper case somewhere on it. The controller will see the bit pattern of the "T" as an ID mark and attempt to resynchronize at that point. The result will always be a bad read. However, if the capital "T" is changed to lowercase, i.e., "tRSDOS,"

the FDC will NOT attempt to resynchronize and the track will be read into memory correctly.

As long as the formatting information of the track image in memory is not disturbed, the track image can be placed back on disk with the WRITE FORMAT TRACK option of the FORMAT utilities. If the formatting information (gap bytes, ID marks, header fields, etc.) are changed, writing the image back to disk may result in an unreadable track.

# CHAPTER 8 - FILE UTILITIES

The final group of routines available in Super Utility 4 are the FILE Utilities, selected by pressing 7 at the main menu. These routines operate very much like the ZAP utilities, except that they are file oriented rather than disk-oriented. This means that if you display a file using the FILE Utilities, you will be able to view all of that file even though it may be in two or more segments located at different places on the disk. The FILE Utilities use the file information maintained in the diskette's directory to know where to find the file. Of course, this also means that you cannot use the FILE Utilities to examine a file on a disk with a non-standard or missing directory.

The available utilities in this group are as shown on the menu:

```
. ** Super Utility 4 ** PowerSOFT Prod. ** By: Kim Watt **  .
.  Copyright # 1984 Breeze/QSD, Inc. - All Rights Reserved  .
. __ File Utilities __                                      .
.                                                           .
.  1 _ Display File Sectors      8 _ Drive Status           .
.  2 _ Compare Files             9 _ Sector Allocation      .
.  3 _ Copy Files               10 _ Build File             .
.  4 _ Disk Directory           11 _ Clear File             .
.  5 _ Free Space               12 _ Disk Allocation        .
.  6 _ Offset File              13 _ Compute Hash Code      .
.  7 _ File Locations           14 _ Compute Passwords      .
.                                                           .
. Choice? #__                                               .
```

## I. DISPLAY FILE SECTORS

This option requires the input of the filename that you wish to see. Do not forget extensions and drive numbers when entering the specification. Super Utility 4 will search its directory when you press ENTER. If it finds the file, it will display the following information:

**END OF ALLOCATION SECTOR (EOA)** - Since a standard DOS allocates disk space in granules, it often happens that a file will not completely fill up the last granule allocated, so that one or two sectors are allocated to it but are actually empty. The EOA number displayed by Super Utility 4 refers to the last sector of the last granule allocated to that file by the operating system.

**END OF FILE SECTOR NUMBER (EOF)** - This is the LAST sector of the file which actually contains file data. It may or may not agree with the EOA number, but will normally be within the last allocated granule.

Super Utility 4 will then prompt you for your choice. This refers to your choice of which sector in the file to view. You need to enter your choice in relative sector form, that is, the first sector of the file is Sector 0, and so on. If you wish to begin viewing the file from the first sector, merely press ENTER; otherwise enter the sector number you wish to see.

Once the display is on the screen, then you may move through the file by using other keys. The arrow keys will page through the file as usual. In addition, you may go to the EOF sector by pressing "E" or to the EOA sector by pressing "A". Pressing the CLEAR key will allow you to select another sector to view.

## TABLE 8-1 - FILE UTILITIES PAGING CONTROLS

| Key | Action |
|---|---|
| **Right or Up arrow** | pages to the next higher sector of the file |
| **Left or Down arrow** | pages to the next lower sector of the file |
| **Sh-right/Sh-up arrow** | displays last allocated sector of the file |
| **Sh-left/Sh-down arrow** | displays initial sector of the file |
| **CLEAR** | requests new relative sector to display |
| **E** | displays end-of-file sector |
| **BREAK** | Returns to File Utilities menu |
| **Shift-BREAK (F1)** | Returns to main menu |
| **@** | Enables DECRYPT mode (see chapter 2). You may need to press SHIFT-@ depending on the CASE setting of the keyboard (toggled with shift-0). |
| **H,D,B,O,Q,A** | Sets modification mode base to hexadecimal, decimal, binary, octal (O and Q are the same) or ASCII respectively |

The arrow keys are used to page through the file. The right and left arrows or the up and down arrows may be used (in this routine the up and down arrows perform the same function as the right and left arrows) to go through the file

one sector at a time. If you attempt to page beyond the limits of the file, you will be given an error message.

At any time, you may enter Modification Mode by selecting your input base (H for hexadecimal, D for decimal, B for binary, O or Q for Octal, or A for ASCII) and then pressing M. All the modification controls available in the DISPLAY DISK SECTORS routine of the ZAP utilities are available to you. See Table 2-2.

The screen display of DISPLAY FILE SECTORS is very similar to that of the DISPLAY DISK SECTORS; however, the leftmost column of information is different, as you can see.

```
   M00#2837 FE05 281E FE06 281E FE08 281E FE0A#(7#.(.#.(.#.(.#.
HEX 410#2812 FE0C 2812 FE11 2812 3E50 3230 BECB#(.#.(.#.(.)P20##
DRV P20#8EF1 E1C9 3E4E 1806 3E45 1802 3E51 3230#####)N..)E..)Q20
 0  R30#8ECB 0102 038B FEF1 F5FE 0A30 053E 0E32###..######.0.).2
TRK 040#2EBE 212B BEC3 77BB CDBB BAD8 0557 3A3A#.#!+##w######W::
 32 W50#BECB 677A 2860 FE61 3809 FE7B 3058 C63F###gz(`#a8.#{0X#?
TRU R60#5718 53FE 2120 0218 4AFE 2220 0218 44FE#W.S#! ..J#" ..D#
 32 /70#2328 40FE 2420 043E C418 3AFE 2520 043E##(@#$ .)#.:#% .)
SEC C80#C518 32FE 2620 043E C618 2AFE 2720 043E##.2#& .)#.##' .)
 13 T90#C718 22FE 2820 043E CB18 1AFE 2920 043E##."#( .)#..#) .)
STO LA0#CD18 12FE 2A20 043E CE18 0AFE 4020 073E##..## .)#..#@ .)
DDD  B0#CF18 02C6 9957 FE20 2014 3A3A BECB 7F20##..##W#  .::##
FPDE C0#0DC5 3AD5 BA4F 0600 CD85 BCC1 1818 7A03#.#:##0..####..z#
RSEC D0#F800 FE0D 200A 3A35 BBB7 2804 3E0A 18EF##.#. .:5##(.)..#
0001 E0#CDDD B020 FBAF D1C9 C5E5 0560 693A D5BA#### ########`i:##
+00 -F0#4F06 007C B528 3CED 4230 0609 552E 0018#0..¦#(<#B0..U...
```

The leftmost column of information first gives you the current modification mode base. Then it displays the drive number, the track number, and the sector number being displayed. Then it displays the data address mark type that it

found on the disk (see chapter 2 for details) along with the density of the disk -- 0SD for single-density, 0DD for double-density. The first character will be either 0 or 1, indicating which side of a disk you are viewing. At the bottom, you will see three lines that look like this:

FPDE
RSEC
0000

**FPDE** stands for "File Primary Directory Entry." Here it means that the sector you are viewing is allocated in the file's primary directory entry. If it said **FXDE**, then the sector you are viewing is allocated in the one of the file's extended directory entries. TRSDOS 1.3 (Model III) does not use FXDE's, so this message should never appear if you are scanning a disk formatted by this particular system.

RSEC stands for Relative Sector. The value then displayed on the third line is the relative sector being displayed as taken from the directory information for that file.

Next to this column is a single vertical column which displays the NAME of the file being viewed, for example,

B
A
S
I
C
/
C
M
D

Toward the bottom of this column, you will see one of three symbols: -, +, or @. The minus sign ("-") indicates that you have not yet reached the EOF sector of the file; The plus sign ("+") indicates that you have already passed the EOF sector

and are viewing a sector that was allocated to the file but not used by it. The @ symbol indicates that you are viewing the EOF sector itself. When this symbol appears, there will be a hex number above it, viewed vertically. This is the first available byte after the end of the file itself. The last byte of the file would be the one preceding this byte.

If this number is 00, it means that the last byte of the file was at relative byte FFH of the preceding sector.

TRSDOS 1.3 and 2.7DD system files need to be handled slightly differently in order to be viewed. This is due to the lack of a standard directory entry for the system files. If you want to view a system file on either of these two systems, for example SYS5, and the disk is in drive 1, reply to the prompt as follows:

**FILENAME ? *05:1**

If the file is inactive in the directory (that is, it has been killed), a "File not found" message will be issued. This format will work only for DISPLAY FILE SECTORS and only for TRSDOS 1.3 and 2.7DD system disks.

If a disk error is encountered, Super Utility 4 will display a message describing the error and give you the option to retry the I/O operation with the prompt,

**R>etry, S>kip, C>ontinuous, N>onstop or Q>uit ?**

Pressing R will cause the program to retry the I/O operation. If the error was due to a momentary condition, this is usually sufficient to correct the situation. If the error appears again, you may select the Continuous or Nonstop options, which will force Super Utility 4 to keep trying to read the bad sector until it gets it right or you stop the process. The only way to escape this is a successful I/O operation or by pressing CLEAR, BREAK, SHIFT-BREAK, F1 or F3.

If you press S, for Skip, Super Utility 4 will go immediately to the sector display routine, with whatever it was able to read before the error forced it to stop. If it was unable to read anything, the display will show all 00's. If it was successful in doing a partial read of the sector, then whatever it was able to read will be displayed. Note that this may not be the same as what is actually on the disk. An error of any kind should always alert you to the fact that the data in Super Utility 4's buffers may be unreliable.

Pressing Q for Quit will abort the entire operation and return you directly to the File Utilities menu.

## II. COMPARE FILES

This routine will allow you to compare two files byte by byte and see if any mismatches exist. You may find this routine useful if, for example, you have doubts about the integrity of a particular file and wish to check it against another copy. You will be asked to enter the source filename (don't forget the drive number!) and the compare filename. Super Utility 4 will scan the two files and report any mismatches on your screen:

**MISMATCH, RELATIVE SECTOR 0000H, BYTE CFH**

At the end of the scan, you will be told how many disk errors (due to CRC errors, etc) were encountered, if any, and the total number of sectors in which mismatches were found between the two files.

Super Utility 4 will compare files up through the EOA (end-of-allocation) sector. This means that two files may show a number of mismatches toward the end. The reason for this is that the contents of sectors beyond the EOF sector is indeterminate. If this is of concern to you, use DISPLAY FILE SECTORS to verify that the differences appear only after the EOF sector.

Also, if the source file for the comparison is shorter than the destination file, but matches the destination file through that point, no mismatches will be reported. If you suspect this is the case, reverse the order of the comparison, i.e., make the destination file the source file and vice versa. Then run the file comparison again. Super Utility 4 will report an error if it encounters the end of the destination file before the source file.

## III. COPY FILES

This routine is a very fast multiple-file copy utility which will let you move any number of files between two disks, the only restraint being the available space on the destination disk. You are asked to supply the source drive number. Super Utility 4 will then read the directory of the diskette in the source drive and display a directory listing on the screen. At the bottom of the screen you will see a filename, followed by its length in grans and sectors, followed by the query, **COPY ?**

If you press "Y", the filename will scroll upward and you will be presented with another filename. If you press "N" then the filename will be replaced with another one without scrolling. If you hold down either key, it will repeat until all the files have been "tagged" for copy or not.

When the last file has been "tagged," you will be asked for the destination drive. Super Utility 4 will scan that drive and display the number of sectors to be copied (the total number of sectors occupied by all the files tagged for copy) and the number of sectors available on the destination disk. It will then proceed to copy the files over one by one. You may specify the same disk drive for destination as for source, and you will be prompted for disk swaps as needed. However, make sure that both source and destination disks are the same DOS type. This is mandatory for single-drive copy

operations.

If a file already exists on the destination disk, Super Utility 4 will display a line of asterisks, " ***** " beside the filename as it copies to indicate that that file is being overwritten. If there is not enough space on the destination disk to hold all the files, the routine will display a message to that effect before starting the copy.

After the last file has been copied over, the directory of the destination disk is updated and written back out to the disk.

This routine will place the files starting at the lowest available track/sector on the destination disk and building upward from there, without leaving any gaps between files. It is a very good routine to use if you wish to "pack" a disk. Where possible, the file copy routine will keep the file in just one extent rather than splitting it up.

If you are copying a file from a TRSDOS 1.3 or 2.7DD formatted disk to a formatted disk created by another system or vice versa, Super Utility 4 will deliberately strip any passwords which may have been set for the source file. The reason for this is that Model III TRSDOS 1.3 computes passwords differently from other operating systems and if the encoded passwords were copied over, even if you gave the correct password, it would not be recognized. If you have any password-protected files on your source disk, you must reassign the passwords using the ATTRIB command of the operating system on the destination disk after the copy routine is completed, or use Super Utility 4's CHANGE FILE PARAMETERS procedure.

Copying a file to a TRSDOS 1.3 or 2.7DD disk from some other system will result in garbage being displayed for the file's creation date when a DIR is done under these two systems. The reason for this is the fact that TRSDOS 1.3 and 2.7DD, unlike all other systems which maintain file dates, do

not maintain the dates in a bit-coded fashion, rather they reserve one byte for the month and one byte for the year. These bytes usually contain some other information, and is not changed by Super Utility 4 when it constructs the destination file entry; hence garbage may appear for the date. If you wish to change the date on these, display the directory entry for the file and modify relative bytes 1 and 2 of the directory entry. Relative byte 1 contains the month in hexadecimal, and relative byte two contains the two-digit year in hexadecimal.

If you press **BREAK** during the copy process, you will be returned to the File Utilities menu, and the destination directory will not be updated.

Due to differences in directory structure, Super Utility 4's COPY FILES will not work if either the source or the destination disk is has a NEWDOS/80 V.2-style extended directory. If you attempt to run this procedure with a disk containing an extended directory as either the source or destination, Super Utility 4 will display the message, "Cannot copy to or from long directories!" and abort.

## IV. DISK DIRECTORY

This routine will display the directory of the disk(s) specified. You will be asked for the drive number. You may specify more than one, separating each number with commas or spaces. Super Utility 4 will scan the disks in order and display for each one the disk name, date, number of formatted tracks, number of free granules, and number of free directory slots. It will then display all the valid files on the disk, along with commas or spaces. Super Utility 4 will scan the disks in order and display for each one the disk name, date, number of formatted tracks, number of free granules, and number of free directory slots. It will then display all the valid files on the disk, along with their file attributes and protection levels. For example, **SIP=7** means a file is a system file, is invisible, and has a protection level of 7.

## V. FREE SPACE

Selection of this option will cause Super Utility 4 to scan all mounted disks and display the disk name, date, number of formatted tracks, number of free granules, free space in Kilobytes and number of free directory slots for each disk. The mounted disks must all be readable, that is, they must have been formatted by a TRSDOS or TRSDOS-compatible system (Model I/III/4/4P) and must contain recognizable directory cylinders. If a non-standard disk is on any drive, you will be presented with an error message.

## VI. OFFSET FILE

This routine will allow you to load a file into memory at one location and cause it to relocate and execute from another location. You may even set up the file so that the TRS-80 interrupts are disabled prior to execution. The file must be in load file format. BASIC program files cannot be relocated.

This routine is useful for making executable files of those programs which normally destroy part of the DOS when they load into memory. Generally, programs written for tape-based systems do this, since they load on top of the DOS resident module. You may move a tape-based (SYSTEM-type) file to disk using the tape-to-disk utilities of your disk system, then use this option to move the file so that it does not destroy the DOS until it is safely loaded into memory. Once the program is in memory, it can be relocated to its normal addresses for execution.

Super Utility 4 will first ask you to supply the filename to be offset. It will then scan the file on disk and tell you the load module range of that file, that is, the present starting address, ending address and transfer (or execution) address of that file. If the file you specified is not an executable load module, the procedure will abort with an error

message.

You will then be asked to supply the new load address, i.e., where you want it to reside when loaded from disk. You will then be asked if you want to add the block move APPENDAGE to the file. This is a short routine which will cause the file to be block-moved to its normal execution addresses after loading into memory. Normally, machine-language files must reside in a particular place in memory in order to execute properly, so if you want the file to execute, you will want to add this appendage.

If you elect to have the appendage added to your relocated file, Super Utility 4 will check to see that there is enough disk space allocated to the file to accommodate it. More disk space will not be allocated to a file by this routine. It will display an error message if insufficient room exists for the appendage routine.

You will then be asked if you want the interrupts disabled or not. Some programs will not execute properly unless the interrupts are disabled. You will have to determine whether the program you are offsetting will work correctly with the interrupts enabled or disabled, and set this option accordingly.

When you press ENTER to complete this last prompt, the file will be written back to disk in its relocated form, and from then on will always load into the new locations when executed from DOS READY level. If you used the block move appendage, it will preserve the contents of the Z-80 registers at load time so that if your program requires them, it will still function correctly.

## VII. FILE LOCATIONS

This option will display complete directory information about each file on a designated disk. You will be asked to enter the drive number to be scanned, and you may enter more than one drive number. Each drive specified must contain a standard formatted disk with a readable directory track. The disks will be scanned one by one, and information about each file in the directory will be displayed, once file at a time. The screen dump will pause to give you a chance to examine the information; press ENTER to display the next file. There is a brief delay built into the keyboard scan for this routine, so that holding the ENTER key down will display the files at a fast, but not overly fast, rate.

For each active file on each disk, five lines of information will be displayed. For example,

```
SYS1/SYS                SIP=7
FPDE-:0,TRACK=20,SECTOR=05,BYTE=00H,DEC=03H
EOFS = 00005, EOFB = 00H, LRL = 0
EACC = C352H, ELPD = C220H, GRANS = 01
EXTENTS / 19,00,00006 / EOF
```

The first line gives the file's name and attributes. In this example, the file is SYS1/SYS and has the attributes **SIP-7**, meaning it is a system file, invisible, with a protection level of 7 ("no access").

The second line gives information about the file's directory entry. **FPDE** stands for "File Primary Directory Entry," and this entry is located on drive 0, track 20 (the directory track), sector 5, starting at relative byte 00H.

If you were to display this sector on the screen using DISPLAY DISK SECTORS, you would see an entry for SYS1/SYS at that particular location.

DEC stands for "Directory Entry Code," and is the relative byte position of the file's "hash code" in the HIT table. In this example, DEC =03 would mean that relative byte 03 in the HIT sector of the directory contains the "hash code" for SYS1/SYS. The DEC's position in this table is relevant to the position of the FPDE, and this table is used by the DOS to locate files in the directory at high speed, without the need to search through the directory sectors one by one.

The third line gives information about the file's **EOFS** (end of file sector) and **EOFB** (end of file byte). The EOFS value is the last sector used by the file (not necessarily the last sector allocated to the file) and the EOFB is the last byte of the file within the EOFS.

The fourth display line gives the encoded values of the passwords. **EACC** stands for Encoded ACCESS password. It is followed by the two byte hash code of the file's user password, if any. **EUPD** is the Encoded UPDATE password and is the two-byte hash code of the file's owner password. Note that the bytes are displayed in LSB/MSB order, which is the way that you would see them on the disk if you viewed the directory using Disk Zap. For example, the hash of a password may be 42E0H, but would appear on the screen as E042H, which is also they way it would appear if you viewed the directory records directly.

Finally, the number of grans occupied by the file is given on this line.

The last line gives the actual location of the file on the disk. The first number is the track where the file is located, the second number is the starting sector (in this case, sector 0) and the third is the length of the file (actually the number of allocated sectors).

On a standard DOS directory, an exceptionally large file may require an extra directory entry, known as the **FXDE** or File Extended Directory Entry. This is very similar to the

FPDE except that the filename is not contained in it, and it is not displayed when a directory is requested from DOS or Super Utility 4. If such a file was encountered by Super Utility 4 , additional information would have been displayed, giving the FXDE's directory entry code and extents.

TRSDOS 1.3 does not allow FXDE's, so this display should never appear when a TRSDOS 1.3-formatted disk was being scanned.

## VIII. DRIVE STATUS

This option will cause Super Utility 4 to check the status of all active drives in your system and report back on each. Drives which are disabled from the configuration tables, or not powered, will be reported as **NOT IN SYSTEM**. Drives with no disks but otherwise powered up will be reported as such.

This procedure cannot properly detect the "no diskette" status on some of the newer microprocessor-controlled drives due to differences in status signals being returned by the drive unit. Such drives may be reported by Super Utility 4 as **NOT IN SYSTEM**. You should be aware that this will cause that drive to be DISABLED in the configuration table. You will have to return to the configuration system and re-enable the drive before you can use it for anything else.

If drives are found to be NOT IN SYSTEM, the settings for them will automatically be changed in the configuration table to reflect this fact. Thus if you had a drive turned off when you executed this procedure and later turn it on in order to use it, you must return to the configuration table and restore it to an active status by changing the active/inactive indicator for that drive (the plus or minus sign in front of the drive number -- see Chapter 1 for full details on configuring).

## IX. SECTOR ALLOCATION

This option will let you enter a track and sector number on a mounted disk and will report which file that particular sector is assigned to, if any. You will be asked to supply the drive number, track number, and sector number. The disk to be scanned must contain a readable directory track, since this routine will use the directory information to determine whether the specified sector is assigned to any active file. If the specified sector was assigned to a file that has been killed, it will be reported as unassigned.

## X. BUILD FILE

This routine will allow you to create and pre-allocate space on a disk for a file in as contiguous a manner as possible. The pre-allocated space will be noted in the file's primary directory entry, so that when you write to this file later on, using your DOS, you will do so faster since the DOS will not need to keep returning to the directory to find space for a new granule every so often.

You will be asked to supply the filename along with the drive number in standard TRSDOS filespec format. Super Utility 4 will then scan the disk and report back the disk's name, date, number of formatted tracks, number of free granules and free directory slots. It will then ask you to enter the number of granules you wish allocated to this new file. When you press ENTER, the information will be written into the disk's directory.

## XI. CLEAR FILE

This routine will request you to enter the filename of a pre-existing file on a disk. It will permit you to remove all the data from that file without actually removing or killing the file from the directory.

You will be asked, "Are you SURE you want to clear it ?" This will give you a chance to change your mind. Enter "Y" to proceed with the operation, otherwise enter "N". If you press "Y", Super Utility 4 will zero out the data in that file. The file will still be present in the directory but it will in effect be empty.

Be careful in using this routine, as there is absolutely NO way of recovering data from a file that has been cleared.

## XII. DISK ALLOCATION

This routine will display a disk allocation map of the disk in the specified drive. The tracks will be listed in the leftmost column. To the right of each track will be slots for each granule, with one of four possible symbols: the letter "X", a "." an underscore, or the letter L. A granule slot with the letter X indicates that that granule is in use, that is, assigned to an active file. A slot with a period indicates that this granule is available for use. If an "L" appears, then that granule was locked out by the DOS during the formatting process, perhaps due to a flaw in the disk, and is not available. Underscore characters will appear in those slots beyond the disk boundaries, that is, beyond the last formatted track on the disk.

The letter "D" will appear on those granules which are part of the disk's directory track.

## XIII. COMPUTE HASH CODE

This routine will calculate the one-byte HIT table hash code for any filespec. You will be asked to supply the filespec (passwords and drive numbers are not required) and the routine will return a one byte hash code in hexadecimal. This code is what is used by the operating system in the Hash Index Table of the directory.

When the operating system must look up a file, it first calculates this code, then goes to the HIT table to locate it. If this code is found in the HIT table, then its position in the table will correspond to the position of the main directory entry in the following sectors. The system is then able to quickly locate files this way, without having to scan through the entire directory each time.

The hash codes produced by this routine, however, are not unique. That is, two or more filespecs, though different, can generate the same code. This is known as a "collision." The system handles the collision by first checking with the main directory entry and comparing it with the user-supplied filename. If the two do not match, then the system goes back to the HIT table to continue the search.

## XIV. COMPUTE PASSWORDS

This routine will allow you to either encode or decode passwords using the algorithms employed by the DOS. When this option is selected, you will be asked whether you want to Encode or Decode a password. To encode a password, simply enter E. You will be asked for the password to encode. Type it in, and the routine will return the 2-byte hash of that password. The algorithm used by this routine will depend on the configuration of drive 0. If drive 0 is configured for TRSDOS 1.3 (T3D) then the TRSDOS 1.3 algorithm will be used; if drive 0 is configured to T1D, then the TRSDOS 2.7DD algorithm will be used. For all others the standard algorithm will be used. You may have to enter the configuration routine to all the settings for drive 0 to get a correct encode of your password. There are no overrides for the compute passwords routine.

To decode a password, enter D. You will be asked for a filename. This filename must exist on one of the mounted disks. When you enter the filename, Super Utility 4 will read

that file's directory entry into memory. It will then display the two hexadecimal bytes which make up the ACCESS (user) password. At this time you will see a furiously-changing graphics character at the bottom of the screen. This indicates that Super Utility 4 is in the process of decoding that password. When it finds a password which translates into the correct hash code, it will display it, and go to work on the UPDATE (owner) password. Again the graphics character will appear. When it successfully decodes the update password, it will display it.

If no password is displayed for either the update or access passwords, it usually means that the two-byte hash is a result of BLANK characters, meaning there is NO password.

Pressing CLEAR while the graphics character is on the screen will abort the operation and proceed to the next step.

With the exception of TRSDOS 1.3 and 2.7DD, all disk operating systems use the same password encoding algorithm. TRSDOS 1.3 and 2.7DD, however, each use a different algorithm. This means that a password which encodes one way on other systems will not encode the same way on these two systems. Also, the TRSDOS 1.3 algorithm is written in such a way that there is one uncodable value -- 0000H. While certain combinations of characters will produce a 0000H hash encode under TRSDOS 1.3, TRSDOS automatically increments the value by 1 when this happens. The 0000 byte pair is used as a protection scheme on certain TRSDOS files, to prevent access.

This difference can produce certain problems, most notably that when files are copied from one system to the other, formerly valid passwords suddenly stop working, and files which had no passwords at all suddenly acquire them. Super Utility 4 deals with this problem by striipping the passwords from the files after the transfer if one of the disks involved is either TRSDOS 1.3 or 2.7DD.

This page intentionally left blank

# CHAPTER 9 - MESSAGES

Below is a list of the messages that Super Utility 4 may generate at various points, along with a brief explanation of each. Users should make careful note of the wording of some of the messages, to prevent possible confusion. For example, Super Utility uses a phrase like "nn sectors not copied" to indicate any unsuccessful sector copy operations. If the operation was completely successful, however, the phrase will appear as "00 sectors not copied," indicating that all sectors involved were successfully copied. Don't let the double-negative phrasing throw you.

*********

During a file copy operation, any file on the destination disk which has the same name as a file being copied over will be overwritten with the new data.

**A>LLOCATION TABLE OR E>NTIRE SECTOR?**

During a GAT repair, Super Utility 4 may repair only the track allocation table, or alternatively, the entire sector including the disk name, date, auto command, etc. If you specify E for entire sector, Super Utility 4 will insert its own data for disk name, date, etc.

**ADD APPENDAGE?**

Prompt to the user to specify whether or not the block move appendage is to be added to a file that has been offset from its normal load addresses. Reply "Y" or "N".

**BAD BACKWARD LINK !**

One or more of a file's Extended Directory Entries (FXDE) has been found to not point back to the preceding FXDE (or FPDE).

**BAD EXTENTS !**

A file has been found to have incorrect extents in the directory entry.

**CANNOT COPY TO/FROM LONG DIRECTORIES**

An attempt was made to use the COPY FILES procedure when one or both of the disks involved had an extended directory (NEWDOS/80).

**CANNOT ESTABLISH DISK TYPE**

Indicates that Super Utility 4 is not able to recognize a disk as being formatted by one of the valid DOS types.

**CANNOT LOCATE DIRECTORY ON DRIVE n**

Super Utility 4 could not read the directory on the specified drive. Usually indicates that the directory track has been written with incorrect or incompatible data address marks.

**CANNOT LOCATE DIRECTORY! TRACK?**

The program could not locate the directory and is requesting the user to specify its location.

**CANNOT UPDATE DIRECTORY**

An attempt to update a diskette directory failed for some reason.

**CANNOT WRITE DIRECTORY !**

During any I/O operation which requires the updating of the diskette's directory, Super Utility 4 was unable to do the update. This may occur, for example, if the target disk is write-protected.

**DATA CRC ERROR**

The CRC bytes for the sector data were wrong. This may have several causes: the sector is a "protected" sector, or the disk drive may be at fault. If it is intermittent, suspect a drive fault (e.g., a worn head pressure pad).

**DATA LOST**

Data was lost during a sector read/write operation. Usually due to software problems (the disk transfer code of the program was not fast enough to keep up with the floppy disk controller).

**DATA LOST ON TRACK READ**

While reading a track of data from a disk, some event occurred which caused all or part of the track to be lost. Possible causes are a slow CPU, or excessively high drive speed.

**DATA LOST ON TRACK WRITE**

While writing data to a track, the timing was off by a sufficient amount so that data was lost before it could be written. This may happen if the TRS-80 CPU has been slowed down.

**DESTINATION ?**

Prompt for the destination drive or drives for a file copy or backup operation. You may enter more than one destination drive, separated by commas or spaces.

**DIRECTORY THERE !**

An attempt was made to move a track to the place occupied by the diskette directory.

**DIRECTORY UNREADABLE**

The directory is non-standard, or damaged.

**nn DISK ERRORS**
**nn SECTOR MISMATCHES**

Reports the total number of disk I/O errors encountered, and the total number of sectors in which mismatches were found, upon completion of a file compare or sector compare operation.

**DISK READ ERROR**

An attempt to read a disk failed.

**DISK WRITE ERROR**

An attempt to write to a disk failed.

**DRIVE DROPPED READY**

A selected drive dropped its "ready" status bit before the I/O operation could be completed.

**DRIVE TIME OUT !**

The drive shut down before the I/O operation could be completed.

**DRIVE n DE-ACTIVATED**

The specified drive has been disabled in the Super Utility 4 configuration table. Change the +/- setting for this drive to bring it into the system.

**DRIVE n MOUNTED AND READY**

Message returned by the Drive Status routine for all drives which are powered on and have diskettes mounted.

**DRIVE n NOT IN SYSTEM**

Message returned by the Drive Status routine for all drives which are either not physically present, not powered, or disabled in Super Utility 4's internal configuration tables.

**EOF DEST FILE REACHED !**

During a file compare operation, the end of the destination file was encountered unexpectedly.

**ERROR ON TRACK WRITE**

An error occurred during an attempt to write memory to a disk track. May indicate a hardware problem.

**FILE ALREADY EXISTS !**

An attempt was made to build a file with a name that already is in the diskette directory.

**FILE IS NOT IN LOAD FORMAT !**

An attempt was made to offset a file that is not in the correct load module format or was not a machine-language file.

**FILE NOT FOUND!**

The specified filespec was not on the disk directory.

**GAT TABLE IS BAD !**

The Granule Allocation Table (GAT) has been found to be damaged or incorrect by the Check Directory routine.

**GRANS TO ALLOCATE ?**

Prompt to the user to specify the size of the file being built. Enter the number of grans that the file is to be allocated.

**nn GRANULES LOCKED OUT**

Reports the total number of unusable granules on a disk following a format operation.

**HARDWARE WRITE FAULT**

An attempt to write to a disk failed due to a defect in the drive or controller hardware.

**HIT TABLE IS BAD !**

The Hash Index Table (HIT) sector has been found to be damaged or contain errors by the Directory Check routine.

**ID CRC ERROR**

The CRC bytes for the track and sector ID fields were wrong. Usually indicates a flawed format.

**INTERRUPT ON PENDING COMMAND**

A disk I/O operation was interrupted while a command to the floppy disk controller was still pending.

**INVALID FILESPEC !**

The user entered a filespec in invalid or non-TRSDOS

format. The correct format for a TRSDOS-type filespec is "FILENAME/EXT.PASSWORD:D." See your operating system manual for more details.

**MISMATCH, RELATIVE SECTOR nnnnn, BYTE xx**

Indicates the position of a mismatch during a file compare or sector compare operation.

**MOUNT DESTINATION DISKETTES**

Prompt to mount the diskettes in all the specified destination drives. Press ENTER to proceed with operation.

**NEXT SECTOR OUT OF RANGE, POSITIONED TO SECTOR xxxxx**

During a Display File Sectors operation, the user attempted to page beyond the boundaries of the file being viewed.

**NO DESTINATION DRIVES !**

No destination drives were specified for a file or sector copy or backup operation.

**NO DISKETTE IN DRIVE n**

Self-explanatory

**NO SPACE AVAILABLE !**

An attempt was made to BUILD a file larger than the available space on a diskette, or to copy files onto a diskette with no free space remaining.

**NON-STANDARD FORMAT**

An attempt was made to read a disk that does not have a TRSDOS-compatible format.

**OPEN DOOR ON DRIVE n**

Self-explanatory.

**R>ETRY, S>KIP, C>ONTINUOUS, N>ONSTOP OR Q>UIT?**

When a disk I/O error is encountered, in most cases, you

will be presented a chance to retry the operation. This prompt gives you several options. R will retry the operation once; C will retry the operation over and over until it succeeds; N is the same as C but no error messages will be displayed during the process; S will skip the portion of the disk where the I/O error occurred; Q will abort the operation completely and return you to the menu.

**SECTOR NOT ASSIGNED TO ANY FILE**

Message returned when the sector specified in a Sector Allocation scan is not assigned to an active file. It may contain data from a KILLED file, however.

**SECTOR NOT FOUND**

Super Utility 4 attempted to read a sector that either was not there (unformatted) or had a non-standard and unreadable format.

**nn SECTORS COULD NOT BE VERIFIED**

Reports the total number of sectors that could not be verified during a Format or Verify Disk Sectors operation.

**nn SECTORS LOST**

During a copy sectors or move sectors operation, if some reason prevents one or more sectors from being successfully written to, this message will appear at the end of the operation.

**nn SECTORS NOT COPIED**

During a copy sectors operation, several sectors could not be read for some reason or another. This message will usually appear in conjunction with some other error message pinpointing the cause of the failure.

**nn SECTORS NOT LOADED**

Several sectors could not be read. Possible causes are the same as above.

**nn SECTORS NOT WRITTEN**

Sectors could not be written during an I/O operation. This message will usually be accompanied by other error messages pinpointing the cause.

**nn SECTORS NOT ZEROED**

During a "Zero sectors" operation, several sectors could not be written to. Possible causes are a hardware drive fault, or incompatible ("protected") formats on the target sectors.

**SOURCE ?**

Prompt for the source drive for a sector or file copy or a backup operation.

**nn TOTAL ERRORS**

Reports the total number of errors found during a directory check.

**TRACK ALLOCATED !**

An attempt was made to move a directory track to a track already occupied.

**TRACK IN BUFFER FROM xxxxH TO yyyyH**
**HIT <ENTER> TO DISPLAY**

Appears on completion of a successful Track-to-Memory operation. The range of addresses occupied by the track data will be given. Press the ENTER key to view the track data.

**TRACK READ ERROR**

An error occurred during an attempt to read a track into memory.

**UPDATING DIRECTORY TO nnn TRACKS**

This message will appear as Super Utility 4 modifies the disk directory's GAT table to reflect the added space if FORMAT was used to increase the track count on a diskette, or if a backup was performed between two disks with differing track counts.

**WRITE FAULT**

Usually signals a hardware problem with the disk drive's write circuitry.

**WRITE PROTECTED DISK**

The diskette has a write-protect tab on it, or the drive has been declared as write-protected in the configuration tables.

This page intentionally left blank

## APPENDIX A
## Some Common Questions Answered

This section contains some of the most common questions asked by users of Super Utility 4.

**In the table of DOS specifiers, several are given for each DOS type! Which one do I use?**

Whichever one you like. All the items under the Model I or Model III columns for a particular DOS type are equivalent.

**What DOS specifier should I use for a double density LDOS Model I data disk?**

Any one of the Model III specifiers. Double-density LDOS disks (with the exception of those with the SOLE modification) are identical for Model I and III. They are also format compatible with TRSDOS 6 disks.

**Why can't I read a single-density TRSDOS disk directory on my Model III/4/4P?**

The problem lies in a hardware difference between the two computers. The Model I uses a floppy disk controller (FDC) chip that is different from the one found in the Model 4/4P. Model I TRSDOS 2.3 (and NEWDOS 2.1) writes its directory track using an address mark that the Model 4/4P FDC simply cannot read. Therefore, when you use SU4 to try and read the directory of a Model I single-density disk on your Model 4/4P,SU4 is unable to find the directory.

**Is there any way around this?**

Yes, there is. SU4 has a procedure in its REPAIR UTILITIES called "Read-Protect Directory." This procedure will take a directory track and write it back out using the correct address marks. If you perform this procedure on a Model I disk while using your Model 4/4P, you will make its directory track readable to the Model 4/4P FDC, and consequently, to SU4.

You should be aware, however, that doing so will make your disk's directory track unrecognizable to Model I TRSDOS, because it now has the wrong address mark! To restore the disk to its original condition, you must perform the exact same procedure on that disk, but using a Model I computer.

**Why won't SU4 read-protect the directory for me?**

It will, but you must tell it to. SU4 will not do anything to a disk unless you ask for it to be done.

**Will the same problem occur in reading a Model 4 disk on a Model I?**

Not as long as the Model I has a doubler board installed. The doubler board actually uses TWO floppy disk controller chips, one of each type, so all bases are covered.

**Does this problem occur with all Model I disks?**

No. It occurs mainly with TRSDOS 2.3 and NEWDOS 2.1 disks, which use the address mark that the Model III can't read. The newer operating systems such as TRSDOS 6, LDOS, MultiDOS and DOSPLUS now write their directory tracks using an address mark common to both the Model I and Model 4/4P FDC's, so the problem does not arise with disks created by them.

**Is it possible to get SU4 to read a hard disk?**

No. Super Utility 4 is strictly a floppy-only utility, and it doesn't even know a hard drive is connected. To get it to read a hard drive would mean adding more subroutines to handle hard drive I/O, and there is simply no room left in the program. In order to access a hard drive you need to use PowerSOFT's Toolbelt utilities which run under TRSDOS 6.

**I bought Super Utility 4 to repair my bad disks. I don't want to know anything about diskette structure, all I want to do is fix my disks! Why can't you make it easier to do this?**

That's like buying a set of tools to repair your car and then expecting the tools to do the work for you while you sit back and drink a beer. Super Utility comprises a set of tools to let you work on your disks. However, you still have to know

what to do in order to use the tools effectively. Since it is impossible for us to anticipate the exact manner in which your disks were damaged, you have to do the work yourself, and this necessarily means learning something about diskette structure and formatting.

**My lineprinter is an MX-80. I set the SU4 configuration to M for MX, and when I turned on DUAL mode, I got garbage on the paper. What gives?**

Your MX-80 is probably equipped with Graftrax-Plus. This EPSON enhancement removed the TRS-80 block graphics characters from the printer. When you configured SU4, you in effect told it that yot your printer still had the block graphics characters. You must reconfigure SU4 and tell it that your printer CANNOT print graphics (it really can't).

**Why doesn't the COPY FILES procedure move system files? Is this a bug?**

No, it's not a bug, it's a safety precaution. For one thing, if you were copying from one DOS to another, you do NOT want to risk copying the system files! Also, several disk operating systems expect to find the system files at specific locations on the disk, and will malfunction if these files aren't where they're supposed to be. Because of the way SU4 "packs" a disk during COPY FILES, the chances are excellent that any system files copied will NOT land in the right place. This is why COPY FILES will not touch system files.

**Will Super Utility 4 access a CP/M disk?**

Yes and no. We know SU4 will access a Montezuma Micro (Dallas) CP/M 2.2 disk, but that's it. Other CP/M formats may or may not be accessible, but we have no information on them. What can you do with a Montezuma Micro CP/M disk? You can use the ZAP utilities for the most part (configure the drive as T3D) and Special Backup to make a copy of the disk, but that's all. Any attempt to use an SU4 procedure which affects the directory on any CP/M disk is certain to DESTROY that CP/M disk.

**What about CP/M Plus disks?**

Don't try it. CP/M Plus uses 512-byte sectors, and accessing such sectors will only show you the first 256 bytes. If you write such a sector back to disk, you will effectively LOSE the second 256 bytes of the sector, writing garbage there instead. You can try Special Backup to make a copy of a CP/M Plus disk, but results may be unpredictable.

Please understand that if you use SU4 to access CP/M (or CP/M Plus or IBM or Atari or Whatever) disks, YOU ARE ON YOUR OWN. PowerSOFT cannot and will not answer any questions regarding the use of SU4 on non-TRSDOS-format disks. Furthermore, PowerSOFT takes no responsibility whatsoever for any damage or loss arising from such attempts. Super Utility 4 is NOT DESIGNED to handle such disks. That it does so is fortuitous coincidence and should not be taken as an indication of future support.